Chapter: Java Basics

What is Programming language?
Programming language is a means to communicate with the machine. It gives set of instructions to the processor. It consists of keyword, syntax, semantics and translator. These translators are often subdivided into compilers and interpreter. Output of the translators is always binary. It should have some sort of input and some sort of output.

Why is there a need for a software?
It reduces the time to do a processing and it has more accuracy which minimizes the human error. It helps to globalize. The communication is faster.

What is the difference between higher level and lower level languages?
Higher level Language: The languages which are written by the users. Java, C++, C#,html. These are converted to assembly language
Lower level Language: The language which are understandable by the machine, processor or the CPU. MIPs, assembly language. These converts to binary numbers.

What is the difference between the software, language and the framework?
Software is built using a language. Language is a means to communicate with the machine. Framework is a like a high level design or layout to build new software.

What is the difference between Procedural Programming and Object Oriented Programming?
Procedural Programming: It is a step by step sequential approach. It is divided in function, top down approach, contains no access specifiers. Ex: c, FORTRAN, Pascal.
Object-oriented Programming: Data and objects are more important. It is divided into objects which are real time. Lots of human interaction for appropriate solution. It has access specifiers, public, private and protected

Why choose Java?
Java is machine independent. It has huge pool of inbuilt library. It has basic security through access specifiers. It can integrate with many advanced and useful frameworks and environments.

What is a class and an object?
Class: It is a blueprint or a template consists of data and function. A class describes a group of objects with common properties and behavior. Class is a virtual thing.
Objects: are often referred to as instances. Instances of objects are created using constructor and the keyword new. It is a real world entity. Every object has different state and behavior.

What is the difference between the state and the behavior?
All the data pertaining to an object is called the state. Based on the data, some of the functionalities might not work or be available which is known as behavior.

What are member functions and data members in class? What are the naming conventions of member functions and data members as well as for class?
Data Members: are the variables which are defined in the class.
Member functions: are the functions which defined in the class.
The naming convention for the member function is Camel casing. All uppercase means constant.
Ex: additionalResultValue.

What is meant by OOPS?
It is a programming language model organized around objects rather than actions and data rather than logic. It has some building blocks application. Primary building blocks are Inheritance, Encapsulation, Polymorphism and Abstraction.

What are driving features of OOPS concept so important?
Encapsulation: Data hiding, wrapping up the data members and member functions to a thing call class. Access specifiers, modifiers are used as a property. Decoupling and strong coupling between data members and member functions. In general, any data is protected. It do not provide access to data members through member function. Through member functions we can only edit
Ex: From Cellphone: The methods would be the contacts to edit something.

Abstraction: It applies only to methods, declaration of method. If class has one abstract then we need to mention the class as abstract. Example TV: it can be C.R.T, LCD, and LED, OLED. The display methods can be described as abstract.

Inheritance: It is when a class gets properties from parent to child and it will have extra features.
Ex: Regular phone to Smart phone
Single inheritance: Class A to Class B
Multilevel inheritance: Class A to Class B to Class C
Hybrid inheritance: Class A and Class B to Class C
Multiple inheritance: It is not possible which from Class A to Class B and Class C

Polymorphism: Existence into multiple forms. Any methods that has multiple forms. There are two type of polymorphisms. One is compile time polymorphism and the other Runtime polymorphism. Compile time polymorphism does method overloading, where the parameters needs to be mentioned. It is based on the data types and at compilation the methods will be combined. Runtime polymorphism does method overriding. Ex: Printing to change printing method.

What is Aggregation and Composition?

Aggregation is having one object in another class. Car: Music Player, engine

Composition is a special form of aggregation. The child is very much needed. Ex: Car without engine is not possible. Therefore, it is hard binding where car without ac is soft binding.

What is a package?

Package is a logical structure or grouping of classes into a single virtual folder. It contains hierarchy of folders.

Simple Calculator Code: Calculator Class

```
package edu.batch.basic; // PACKAGE DECLARATION

import java.lang.Math; // AVAILABLE BY DEFAULT
import java.util.Date; // EXPLICIT

public final class Calculator { // CLASS DECLARATION
	boolean status; // DATA MEMBER -<accSp> <final/static> <datatype>
				// <variablename>=<defaultvalue>;
	final String NAME = "ABC CALC";

	final int add(int a, int b) { // <accSp> <final/static> <returnType>
					// <methdName>(<args>){...}
		System.out.println(new Date());
		int additionResultValue = a + b; // LOCAL VARIABLE - CAMEL CASING
		return additionResultValue;
	}

	double sqrt(int c) {
		return Math.sqrt(c);
	}

	int subtract(int a, int b) {
		return a - b;
	}

	int multiply(int a, int b) {
		return a * b;
	}

	float divide(int a, int b) {
		return a / b;
	}
}
```

Simple Calculator Code: Test Class

```
package edu.batch.basic;

public class Test {
	public static void main(String[] args) {
		Calculator c = new Calculator(); // OBJECT CREATION
		int a = Integer.valueOf(args[0]); // COMMAND LINE ARGUMENTS
		int b = Integer.valueOf(args[1]);
		int result = c.add(a, b);
		System.out.println("The Sum is : " + result);
		{
			int sum = 25; // CHILD SCOPE - INNER SCOPE
		}
	}
}
```

What are the storage mechanism in Java?

Heap: It is a temporary space, aka cache, which stores all the object level data
Stack: all local variables are stored in stack. It uses push and pop method. In 'bin' all the classes would be generated.

What is JVM, JRE and JDK?

JVM is the java virtual machine and the JDK is java development kit and the framework that comes with java. The libraries utility, math are all part of JDK. JRE is the java runtime environment, JRE is the specific implementation of the JVM and JVM is the abstraction of JRE. JVM is platform independent. JVM is the place where it process the data.

What is a constant?

It means something that does not change. Ex: Final String name= "ABC CALC" Generally naming conventions with all uppercase letters are for constants.

What is the scope of a variable?

After the declaration we can access the variable in between the braces.
.

What are the primitive types?

They are Boolean, char, byte and void
They can also be short, int, float and double
Boxing and unboxing, type casting

AutoBoxing: When a primitive type is converted to object. Ex: Integer objCount= count;
Boxing: When a primitive type is converted to object (Putting into larger type) Ex: Double objlong = new long(count)
Unboxing: It is when an object is converted to primitive type. Ex: long unboxedValue = obj.long.longValue();

Simple Primitive Code

```
package edu.batch.basic;

public class Primitives {
	public static void main(String[] args) {
		int count = 25;

		Integer objCount = count; // AUTO BOXING / BOXING

		Long objLong = new Long(count); // BOXING
		System.out.println(objLong);

		long unboxedValue = objLong.longValue(); // UN BOXING

		int x = -25;

		int a = 10, b = 20;

		a = a + b;
		System.out.println(a);
		float percentage = 65f;

		int intPerc = (int) percentage; // EXPLICIT CASTING / DOWN CASTING
		double dPerc = percentage; // IMPLICIT CASTING / UP CASTING
	}
}
```

What are the operators in Java?

Arithmetic(+,-,*,/) at least two operands
Unary(+,-) only one operand Ex: int x=-15
Assignment(=), Arithmatic(+,-,*,/) & Unary (+,-)
Relational (<,>,<=,>=,==,!=) & Logical (&&,||)
Auto Increment(++) & Decrement(--) : Pre & Post
Bitwise (&, |, ^, ~) & Conditional (?,:)

What are the conditional operators?

Question mark and colon. Question represents the true part and colon represent the false part. It is the representation of if statement.

What are the control statements?

If, If-else, While, do-while, for, for-each, switch, Break & Continue are the control statements.

What is the difference between the post increment and pre increment?

Pre increment is when we first increment and then execute the statement.
Post increment is when we first execute and then increment the statement.

Simple Short Hand Operators Code

```
package edu.batch.basic;

public class ShorthandOpers {
	public static void main(String[] args) {
		int[] marks = { 45, 65, 80, 90, 40, 70 };

		int total = 0;
		// for ( 1;start;end)
		for (int i = 0; i < marks.length; i++) {
			if (i == 2) {
				continue;
			}
			total += marks[i];
		}
		total = 0;
		for (int mark : marks) {
			if (mark < 0) {
				continue;
			}
			if (mark > 100) {
				break;
			}
			total += mark;
		}
		System.out.println(total);
		total /= marks.length;
		System.out.println(total);
	}
}
```

Simple Operators Code

```
package edu.batch.basic;

public class Operators {
	public static void main(String[] args) {
		boolean x = true;
		boolean y = false;
		if (x && y) {
			System.out.println("SUCCESS");
		} else {
			System.out.println("FAILURE");
		}

		int i = 5;
		int j = 9;
		int k = ++i - j--;
		/*
		 * i = i + 1; k = i - j; j = j - 1;
		 */

		int p = 10; // 1010
		int q = 13; // 1101
		int r = p ^ q; // 0111
		System.out.println(r);

		System.out.println(~20);

		float price = 199.99f;
		String res = (price > 100) ? "Too Costly" : "Economy";

		if (price > 100) {
			res = "Too Costly";
		} else {
			res = "Economy";
		}

		System.out.println(res);

		System.out.println(i + " and " + j + " and " + k);
	}
}
```

Simple Access Specifier Code: Car Class

```
package edu.batch.oops;

import java.sql.Time;

/*
 * AccessSp         class       package            children           project
 * public           Y             Y                  Y                  Y
 * private          Y             N                  N                  N
 * protected        Y             Y                  Y                  N
 * default          Y             Y                  N                  N
 *
 */

public class Car {
	protected double mileage;
	private float currentSpeed;
	private float availableFuel;
	protected float tripDistance;
	private Character transmission;
	public Time time;
	String name;

	public void drive() {
		// driving functionality
	}

	public void changeSpeed() {
		// accelarate / decelarate
		// current speed
	}

	public void changeTransmission() {
		// change transmission
		// transmission
	}

	public void onOff() {
		// on or off;
		// current speed
		//
	}

	public void showSignals() {
		// shows indicators
	}
}
```

Simple Access Specifier Code: Car Class Extends

```
package edu.batch.oops1;

import edu.batch.oops.Car;

// public, protected
public class SUV extends Car {
	public void autoStart() {
		mileage = 50000;
		tripDistance = 100;
		//currentSpeed = 60;
	}
}
```

Simple Access Specifier Code: Bank Account Class

```
package edu.batch.oops;

public abstract class BankAccount {

	private Double balance;

	// CONCRETE METHODS
	public void debit(double amt) { // DECLARATION
		balance = balance - amt; // DEFINITION
	}

	public void credit(double amt) {
		balance = balance + amt;
	}

	// ABSTRACT METHODS
	public abstract void calcInterest(); // DECLARATION
}
```

Simple Access Specifier Code: Saving Account Extends

```
package edu.batch.oops;

public class SavingsAccount extends BankAccount {

	@Override
	public void calcInterest() {
		// write some logic to calc interest
	}
}
```

Simple Access Specifier Code: Test Class

```
package edu.batch.oops;

public class Test {
	public static void main(String[] args) {
		// ENCAPSULATION
		Car c = new Car();
		c.drive();
		c.name = "John's Car";

		// ABSTRACTION
		BankAccount account = new SavingsAccount();
		account.calcInterest();
	}
}
```

Simple Inheritance Code

```
package edu.batch.oops;

public class ClassA {
	public void print() {
		System.out.println("CLASS A");
	}
}
```

```
package edu.batch.oops;

public class ClassB {
	public void print() {
		System.out.println("CLASS B");
	}
}
```

```
package edu.batch.oops;

public class ClassC extends ClassA {
	public void print() {
		super.print();
	}

	public static void main(String[] args) {
		ClassC c = new ClassC();
		c.print();
	}
}
```

Simple Polymorphism (Compile Time/Method Overloading) Code

```
package edu.batch.oops;

public class Contact {

	public void create(String name, Long phone) { // 1
		System.out.println("1");
	}

	public void create(String name, String email) { // Type
		System.out.println("2");
	}

	public void create(String name, Long phone, String email) { // count
		System.out.println("3");
	}

	public void create(String name, String email, Long phone) { // order
		System.out.println("4");
	}

	public static void main(String[] args) {
		Contact c = new Contact();
		c.create("John", 8789542255L);
		c.create("John", "john@hotmail.com", 7855252222L);
	}
}
```

How do you implement method overloading and method overriding?

Method overloading have the same method repeating multiple times but their parameters needs to be changed which is based on the order, based on the type or based on the number of arguments. Method Overriding when you can override the parent class arguments.

Simple Polymorphism (Run Time/Method Overriding) Code

```
package edu.batch.oops;

public class Calculator {
	public Integer add(Integer a, Integer b) {
		return a + b;
	}
}
```

```
package edu.batch.oops;

public class AdvanceCalc extends Calculator {
	private Integer multipler = 2;

	@Override
	public Integer add(Integer a, Integer b) {
		return multipler * (a + b);
	}

	public static void main(String[] args) {
		Calculator c;
		if (args[0].equalsIgnoreCase("C")) {
			c = new Calculator();
		} else {
			c = new AdvanceCalc();
		}
		System.out.println(c.add(20, 10));
	}
}
```

What are the constructors and its types?

It is a default set of steps which initializes an object. It also gives default values to data members. Constructors creates an object. When the new keyword is mentioned, this is when the constructor is executed. Constructors have methods which do not return anything. There are three types of constructors: Simple Constructor, Parameterized Constructor and Copy Constructor. Constructor have methods which do not return anything.

Simple Constructor Code: Account

```
package edu.batch.oops1;

import java.util.Random;

public class Account {
	private String name;
	private Long accNo;
	private Double balance;

	// Default is already available if no constructor exists

	// simple constructor
	public Account() {
		System.out.println("Initialising Simple");
		name = "UNKNOWN";
		accNo = 0L;
		balance = 0.0;
	}

	// Paremeterized Constructor
	public Account(String name, Long accNo, Double balance) {
		System.out.println("Initialising Parameterised");
		this.name = name;
		this.accNo = accNo;
		this.balance = balance;
	}

	// Copy Constructor
	public Account(Account a) {
		this.name = a.name;
		this.accNo = a.accNo;
		this.balance = a.balance;
	}

	public Account(String name, double balance) {
		this.name = name;
		this.balance = balance;
		this.accNo = (new Random()).nextLong();
	}

	public void setAccNo(long l) {
		this.accNo = l;
	}
}
```

Simple Constructor Code: Test

```
package edu.batch.oops1;

import edu.batch.oops.Car;

public class Test {
	public static void main(String[] args) {
		Car c = new Car();
		c.drive();
		// c.name = "John's Car";
		c.changeSpeed();

		Account a = new Account();
		Account b = new Account("Adam", 2500.00);
		Account d = new Account(b);
		d.setAccNo(98765431L);
	}
}
```

What is a Static Keyword? Where it is used?

It simple means no motion. Every object has different data member but there is some data which is shared across the object. Ex: All the accounts in the same class (Bank) will have the same BankAccount. It is used in static variables (data members), methods, functions and blocks. They belong to the class, they do not belong to the object.

Simple Static Keyword Code

```
package edu.batch.oops1;

public class StaticKeyword {

	private int x = 0;
	private static int y = 0;

	// executed when the class is loaded into the JVM
	static {
		System.out.println("In Static Block");
		y = 55;
	}

	public static void main(String[] args) {
		StaticKeyword.printY();
		StaticKeyword s1 = new StaticKeyword();
		StaticKeyword s2 = new StaticKeyword();
		StaticKeyword s3 = new StaticKeyword();
		StaticKeyword s4 = new StaticKeyword();

		s1.x = 10;
		s1.y = 20;

		s2.x = 30;
		s2.y = 40;

		s3.x = 50;
		s3.y = 60;

		s4.x = 70;
		s4.y = 80;

		StaticKeyword.y = 100;
		StaticKeyword.printY();

		System.out.println(s1.x + "=" + s1.y); // 10 80
		System.out.println(s2.x + "=" + s2.y); // 30 80
		System.out.println(s3.x + "=" + s3.y); // 50 80
		System.out.println(s4.x + "=" + s4.y); // 70 80
	}

	private static void printY() {
		System.out.println(y);
	}
}
```

What are the differences between abstract class and the interfaces?

Abstract Class: It is a class which contains abstract methods. Abstract method is a method which where declaration exists but no mentioning of definition, implementation exists. Abstract class do not have any constructors. It cannot be instantiated. Multiple inheritance is not supported in the abstract class. Abstract classes are classes and can have concrete methods. Normal method can exist in abstract class. . It can have constructors. You need to mention abstract.

Interface: It is a way to communicate with the underlying entity. Ex: Monitor, T.V, T.V is an object, buttons yes, behavior and operations yes. Therefore, remote is an interface of the television. Interfaces are the interfaces. In interface all the methods are abstract. Interfaces cannot have constructors. All the variable are pubic, static and final. Interfaces are implemented. Interface can implement multiple inheritance but abstract class cannot implement multiple inheritance.

Simple Interface Code: Remote Control

```
package edu.batch.interf;

public interface RemoteControl {
	String uniqueId = "8986Lk5454";

	public abstract void changeVolume();

	void changeChannel();

	public void changeColor();

	public void changeMode();
}
```

Simple Interface Code: TV

```
package edu.batch.interf;

public class TV implements RemoteControl, RemoteApp {

	@Override
	public void changeVolume() {
		System.out.println("changing volume");
	}

	@Override
	public void changeChannel() {
		System.out.println("changing channel");
	}

	@Override
	public void changeColor() {
		System.out.println("changing color");
	}

	@Override
	public void changeMode() {
		System.out.println("changing mode");
	}

	@Override
	public void connect() {
		System.out.println("Connecting");
	}

	@Override
	public void disconnect() {
		System.out.println("Disconnecting");
	}

}
```

Simple Interface Code: Remote App

```
package edu.batch.interf;

public interface RemoteApp {

	public void changeVolume();

	public void connect();

	public void changeColor();

	public void disconnect();
}
```

What is Exception Handling?
Exception is an error event that can happen during the execution of a program and disrupts its normal flow.
Failure: It is also known as fault; Ex: System crash, Facebook (who owns the software), something which is not the problem of the software. Ex: Virus, power failure.
Exception: Ex: Facebook cannot login due to typing wrong login or password, may be database crash. Error and Exception are usually throwable problems class

What is the throwable class?
It is a class in java which is a parent of all errors and exception. Everyone needs to extend from throwable. Exception and error extends from that.

How do you create a user defined custom exception?
Two different ways of creating an exception are extending the exception class or extending the throwable class.

What are the primary keywords for exception handling?
Try, catch, throw, throws and finally

How to manually invoke an exception?
Throw followed by the exceptions

When to use throws keyword?
If your method throws keyword back to the calling main method. The method declaration should have the method followed by the declaration.

What are the difference between final, finale and finalize?
Final Values: It cannot be changed. Final Methods: It cannot be override using runtime polymorphism. Final Classes: It cannot be inherited, cannot be extended.
Finale: The block will get executed even there is exception or not
Finalize: It will get executed before the object is garbage collected.

What is the difference between checked exception and unchecked exception?
Checked exception: This exception needs to be taken care of; usually custom exception. The compiler will throw error if not taken care of.
Uncheck exception: Ex: Arithmetic exception, Null pointer exception, class not found. These are runtime errors.

Simple Exception Code

```
package edu.batch.exception;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class FileOperations {
	public static void main(String[] args) throws IOException {
		FileReader fr = null;
		BufferedReader br = null;
		try {
			fr = new FileReader("C:\\temp\\in.txt");
			br = new BufferedReader(fr);
			int a = getInt(br);
			int b = getInt(br);
			if (b == 0) {
				throw new InvalidInputException(b);
			}
			int c = a / b;
			System.out.println("The result is : " + c);
		} catch (FileNotFoundException e) {
			System.out.println("File does not exist");
		} catch (NumberFormatException e) {
			e.printStackTrace();
		} catch (InvalidInputException e) {
			System.out.println(e.getMessage());
		} finally {
			fr.close();
			br.close();
		}
	}

	private static int getInt(BufferedReader br) throws NumberFormatException,
			IOException {
		try {
			return Integer.parseInt(br.readLine());
		} finally {
			System.out.println("completed parsing");
		}
	}

	public void finalize() {
		System.out.println("destroying the object");
	}
}
```

Simple Exception Code*: *Invalid Input Exception

```
Package edu.batch.exception;

public class InvalidInputException extends Exception {

	private String message;

	public InvalidInputException(int b) {
		message = "Invalid input : " + b;
	}

	public String getMessage() {
		return message;
	}

}
```

What is the Serialization in the java?
When we write the state of an object into an external string and store it in a file. Serializable is a marker interface.

What is the marker interface?
It is an empty interface it does not have any method or no inheritance

What is the deserialization in the java?
Reading from a file and converting that to an object.

What is transient variable?
If any object has transient variable, that variable won't get serialized. Data members won't get serialized.

What is volatile variable?
These are variable which keeps on changing. Its data do not store in the heap. It stores directly in the memory. It will always be read from the main memory

What is Synchronization?
It is when no two threads can execute at the same time. Piece of code needs to be synchronized. One of them needs to wait. Ex: Bank Account shared by two members, their account has $1000. Both of the members goes to the ATM to pick up the money at the same time. The ATM booth would show some balance to both of them (through Database check, not valid transaction). Only one person can have the access to the same account.

Difference between String Builder and String Buffer?
String builder is not synchronized and not thread safe. It is good for single thread.
String Buffer is synchronized and it is thread safe. It can be used for multiple threads.

What is the Immutability Concept?
In strings if two variables or objects has the same value, the word that would be saved are in the same memory location. For integer it does not work. In this way, the strings saves the memory location.

Simple Immutability Code

```
package edu.batch.class3;

public class Immutability {
	public static void main(String[] args) {

		String s = "Sreenath"; // 1000 = Sreenath
		s = "Prakash"; // 2000 = Prakash
		s = "Vinod"; // 3000 = Vinod
		s = s + " Kumar"; // 4000 = Vinod Kumar

		int i = 25; // 100 = 25
		i = 40; // 100 = 40

		String a = "John"; // 5000 = John
		String b = "John"; // 5000 = John

		int k = 25; // 200
		int j = 25; // 300

		// StringBuider = Not Synchronized / Not Thread safe = For Single Thread
		// StringBuffer = Synchronized / Thread Safe = for multi thread
		StringBuilder sb = new StringBuilder("Welcome "); // mutable - 8000
		sb.append("to Java Tutorial.");
		sb.append("This is a presentation on Strinbuffer and Builder"); // 8000

		System.out.println(sb);

	}
}
```

What is Cloning?
To create a copy of an existing object.

Simple Cloning Code: Employee

```
package edu.batch.class3;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

public class Employee implements Cloneable, Serializable {
	private Long id;
	private String name;
	private transient Integer age;

	public void serialize() throws IOException {
		FileOutputStream out = new FileOutputStream("c:\\temp\\emp.obj");
		ObjectOutputStream oos = new ObjectOutputStream(out);
		oos.writeObject(this);
		oos.close();
		out.close();

	}

	public void deserialize() throws IOException, ClassNotFoundException {
		FileInputStream fis = new FileInputStream("c:\\temp\\emp.obj");
		ObjectInputStream ois = new ObjectInputStream(fis);
		Employee e = (Employee) ois.readObject();
		System.out.println(e);
	}

	@Override
	public String toString() {
		return "Employee [hashcode= " + this.hashCode() + ", id=" + id
				+ ", name=" + name + ", age=" + age + "]";
	}

	public Employee(Long id, String name, Integer age) {
		super();
		this.id = id;
		this.name = name;
		this.age = age;
	}

	public Employee clone() {
		Employee e = new Employee(id, name, age);
		return e;
```

```
	}

	public Long getId() {
		return id;
	}

	public void setId(Long id) {
		this.id = id;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public Integer getAge() {
		return age;
	}

	public void setAge(Integer age) {
		this.age = age;
	}

}
```

Simple Cloning Code: Test

```
package edu.batch.class3;

import java.io.IOException;

public class Test {
	public static void main(String[] args) throws IOException,
			ClassNotFoundException {
		Employee e = new Employee(12345L, "Adam", 30);
		Employee e1 = e.clone(); // copy
		Employee e2 = e; // reference
		e2.setAge(40);
		System.out.println(e);
		System.out.println(e1);
		System.out.println(e2);

		e.serialize();
		e.deserialize();
	}
}
```

What is Garbage Collection?
A data that can no more be used. Any non-referenced a memory location/object you cannot reference it back. (JVM cache) Heap memory location. No variable pointing/ unreferenced to a memory location. Main method is usually daemon/ background thread which is nothing but garbage collection thread. Command: system.gc(), it requests JVM for garbage collecting. There are levels of garbage collection, the large object with lot of properties, attributes is based on when, how large, data type, when the object was created.

What is the difference between '/n' and '/r'?
/n means newline and /r means carriage returns. Some of the editors might not understand /n. The properties of both of them are same.

What are the Escape characters?
\n –newline
\r –carriage return
\t –tab
\\-prints backslash
\" –prints "
\' –prints '

Simple Escape Characters Code

```
package edu.batch.class3;

import java.io.FileWriter;
import java.io.IOException;

public class Escape {
	public static void main(String[] args) throws IOException {
		System.out
				.println("Welcome to java training.\n This is a class on
Core Java");

		FileWriter fw = new FileWriter("c:\\temp\\esc.txt");
		fw.write("Welcome to java training.\r This is a class on Core Java");
		fw.close();
	}
}
```

CHAPTER: COLLECTIONS

What is Collections and its criteria?

It depends on lists. Whether we need duplicates or not

It is an object that can hold references to other objects. The collection interfaces declare the operations that can be performed on each type of collection. Therefore the basic properties are: They allow duplicates or not, ordered or not ordered (retrieving in the same order or not), they are homogenous or heterogeneous.

What are the collection interfaces?

It is applicable to list and Set but not with Map. Collections can be Collection: Basic Operators, Collection: Iterators, Collection: Bulk Operations, Collection: Array Operations.

Differentiate b/w List/Set/Map?

List: Ordered with duplicates

Set: Unordered, duplicates are not allowed

Map: key value pairs, key unique, value is same

Collection: Basic Operations

```
int size( );
boolean isEmpty( );
boolean contains(Object element);
boolean add(Object element); // Optional
boolean remove(Object element); // Optional
Iterator iterator( );

public interface Iterator{
booleanhasNext( );
// true if there is another element
Object next( );
// returns the next element (advances the iterator)
void remove( ); // Optional// removes the element returned bynext}

static void printAll(Collection coll)
{Iteratoriter=coll.iterator( ); while (iter.hasNext( ))
{System.out.println(iter.next( ) );}
}

Note that this code is polymorphic--it will work for any collection
```

Collection: Bulk Operations

```
boolean containsAll(Collection c);
boolean addAll(Collection c); // Optional
boolean removeAll(Collection c); // Optional
boolean retainAll(Collection c); // Optional
void clear( ); // Optional
addAll, removeAll, retainAllreturn trueif the object receiving the message was
modified
```

Collection: Array Operations

```
Object[ ] toArray( );
creates a new array ofObjects
Object[ ] toArray(Object a[ ]);
Allows the caller to provide the array
Examples:
Object[ ] a = c.toArray( );
String[ ] a;a = (String[ ]) c.toArray(new String[0]);
```

What are some of the basic operations of Collections?

Iterator collection, adding, adding all the elements, retaining all of the elements, Array2 [], containsAll, size of the collection, bulk and array operations, contains, containsall are some of the basic operations of Collections.

Simple Collection: Basic Operations Code

```
package edu.batch.coll;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

public class BasicOps {
	public static void main(String[] args) {
		Collection<Integer> primes = new ArrayList<Integer>();
		primes.add(2); // 0
		primes.add(3);
		primes.add(5);
		primes.add(7);
		primes.add(11);
		primes.add(13);
		primes.add(17);
		primes.add(18);
		primes.add(19);

		primes.remove(18);

		System.out.println(primes.size());

		System.out.println(primes.contains(17));

		System.out.println(primes.isEmpty());

		Integer sum = 0;
		for (Iterator iterator = primes.iterator(); iterator.hasNext();) {
			Integer val = (Integer) iterator.next();
			sum += val;
			if (val > 10)
				iterator.remove();
		}
		System.out.println(primes);
		System.out.println(sum);
	}
}
```

Simple Collection: Bulk Operations Code

```
package edu.batch.coll;

import java.util.ArrayList;
import java.util.Collection;

public class BulkOps {
	public static void main(String[] args) {
		Collection<Float> prices = new ArrayList<Float>();
		prices.add(9.99f);
		prices.add(99.99f);
		prices.add(19.99f);
		prices.add(19.99f);
		prices.add(9.99f);
		prices.add(19.99f);
		prices.add(29.99f);
		prices.add(49.99f);
		prices.add(79.99f);
		// 9.99, 99.99, 19.99, 19.99, 9.99, 19.99, 29.99, 49.99, 79.99
		Collection<Float> checkList = new ArrayList<Float>();
		checkList.add(99.99f);
		checkList.add(19.99f);
		System.out.println(prices.containsAll(checkList));

		prices.addAll(checkList);
		// 9.99, 99.99, 19.99, 19.99, 9.99, 19.99, 29.99, 49.99, 79.99, 99.99,
		// 19.99
		prices.retainAll(checkList);
		// 9.99, 9.99, 29.99, 49.99, 79.99

		System.out.println(prices);

		prices.clear();
		System.out.println(prices);

	}
}
```

Simple Collection: Array Operations Code

```
package edu.batch.coll;

import java.util.ArrayList;
import java.util.Collection;

public class ArrayOps {
	public static void main(String[] args) {
		Collection<String> names = new ArrayList<String>();
		names.add("Deepak");
		names.add("Sanjay");
		names.add("Anish");
		names.add("Piyush");

		Object[] arr = { "Manish", "Joseph", "Adam", "Prakash", "Madhav",
				"Karl" };

		arr = names.toArray(arr);

		for (int i = 0; i < arr.length; i++) {
			System.out.println(arr[i]);
		}
	}
}
```

What is the List Interface?

```
A list is ordered and may have duplicates
Operations are exactly those for Collections
intsize( );
booleanisEmpty( );
booleancontains(Object e);
booleanadd(Object e);
booleanremove(Object e);
Iteratoriterator( );
booleancontainsAll(Collection c);
booleanaddAll(Collection c); booleanremoveAll(Collection c);
booleanretainAll(Collection c);
void clear( );
Object[ ] toArray( );
Object[ ] toArray(Object a[ ]);
```

What are the two implementations of Lists? What are the differences between them?

Array List and Linked list are the two implementations of Lists. List is an interface; you can't say new List ( ) LinkedListgives faster insertions and deletions.ArrayListgives faster random access. It's poor style to expose the implementation, so:Good: List list= new LinkedList( );Bad: LinkedListlist = new LinkedList( );*ArrayList* uses index buckets. Therefore it searches faster. (also generally total ArrayList is fast).*LinkedList* uses data and address part. Therefore, it is used for insertion or deletion. *Queue* is a collection for holding elements prior to processing.

What is the inherited list methods?
list.remove(e) removes the first e. add and addAll add to the end of the list
To append one list to another: list1.addAll(list2);
To append two lists into a new list: List list3 = new ArrayList(list1);list3.addAll(list2);
Again, it's good style to hide the implementation

What are the additional operations of the Lists?
Add, remove, gettheindex, positionalaxis, last index, sublist are the additional operations of the Lists.

Simple List Code

```
package edu.batch.coll.list;

import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;

public class ListExample {
	public static void main(String[] args) {
		List<Date> holidays = new LinkedList<Date>();

		holidays.add(getDate(12, 01, 2016));
		holidays.add(getDate(12, 01, 2016));
		holidays.add(getDate(15, 1, 2015));
		holidays.add(getDate(02, 11, 2015));
		holidays.add(getDate(01, 01, 2016));
		holidays.add(getDate(22, 01, 2016));
		holidays.add(getDate(12, 01, 2016));

		for (Iterator iterator = holidays.iterator(); iterator.hasNext();) {
			Date date = (Date) iterator.next();
			System.out.println(date);
		}
	}

	private static Date getDate(int day, int month, int year) {
		Calendar c = Calendar.getInstance();
		c.set(Calendar.DAY_OF_MONTH, day);
		c.set(Calendar.MONTH, month - 1);
		c.set(Calendar.YEAR, year);
		return c.getTime();
	}
}
```

What is the List: Positional access?
Object get(intindex); // Required --// the rest are optional
Object set(intindex, Object element);
void add(intindex, Object element);
Object remove(intindex);
abstract booleanaddAll(intindex, Collection c);
These operations are more efficient with theArrayListimplementation

Simple List: Positional Access Code

```
package edu.batch.coll.list;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class PositionalAccess {
	public static void main(String[] args) {
		List<Long> ssns = new LinkedList<Long>();

		ssns.add(123456789L);
		ssns.add(987456321L);
		ssns.add(456782139L);
		ssns.add(951487226L);
		ssns.add(965874522L);

		System.out.println(ssns.get(1));
		System.out.println(ssns);
		ssns.set(3, 968547855L);
		System.out.println(ssns);
		ssns.add(2, 853658458L);
		System.out.println(ssns);

		List<Long> newList = new ArrayList<Long>();
		newList.add(965845875L);
		newList.add(325847196L);

		ssns.addAll(0, newList);
		System.out.println(ssns);

		ssns.remove(ssns.size() - 1);
		System.out.println(ssns);

		System.out.println(ssns.getClass());
	}
}
```

What is the List: Searching?
intindexOf(Object o);
intlastIndexOf(Object o);
equals and hashCode work even if implementations are different

Simple List: Searching Code

```
package edu.batch.coll.list;

import java.util.ArrayList;
import java.util.List;

public class Searching {
	public static void main(String[] args) {
		List<Integer> last4 = new ArrayList<Integer>();

		last4.add(6789);
		last4.add(6321);
		last4.add(2139);
		last4.add(7226);
		last4.add(6321);
		last4.add(9658);
		last4.add(6321);
		last4.add(9584);
		last4.add(6321);

		System.out.println(last4.indexOf(6321));
		System.out.println(last4.lastIndexOf(6321));

		int firstOccurance = last4.indexOf(6321);

		List<Integer> subList = last4.subList(firstOccurance + 1, last4.size());
		System.out.println("Second Occurance : "
				+ (firstOccurance + subList.indexOf(6321) + 1));

		for (int i = 0; i < last4.size(); i++) {
			if (last4.get(i) == 6321) {
				System.out.println(i);
			}
		}

	}
}
```

What are the List iterators?
Iterators specific to Lists:
ListIteratorlistIterator( );
ListIteratorlistIterator(intindex);
starts at the position indicated (0 is first element)
Inherited methods:
booleanhasNext( );
Object next( );
void remove( );
Additional methods:
booleanhasPrevious()
Object previous()

What are the List iterators: Backwards?
booleanhasPrevious( );
Object previous( );
intnextIndex( );
intpreviousIndex( );
Think of the iteratoras "between" elements
Hence, nextfollowed by previous gives you the same element each time

What are more operations of Lists?
void add(Object o);
Inserts an object at the cursor position
Object set(Object o); // Optional
Replace the current element; return the old one
Object remove(intindex); // Optional
Remove and return the element at that position

Simple List: Iterator Code

```
package edu.batch.coll.list;

import java.util.ArrayList;
import java.util.List;
import java.util.ListIterator;

public class ListIteratorExample {
                    public static void main(String[] args) {
                         List<Long> barcodes = new ArrayList<Long>();

                         barcodes.add(548599665566554L);
                         barcodes.add(846454684484645L);
                         barcodes.add(646874645454344L);
                         barcodes.add(984647613333533L);
                         barcodes.add(813164318464466L);
                         barcodes.add(366451684448435L);

                         System.out.println(barcodes);

                         ListIterator<Long> li = barcodes.listIterator();
                         while (li.hasNext()) {
                               System.out.print(li.previousIndex() + " ");
                               System.out.print(li.next() + " ");
                               System.out.println(li.nextIndex());
                               li.set(000000000000000000L);
                               li.add(1L);
                         }
                         System.out.println(barcodes);
                    }
}
```

What are List: Range-View?

List subList(int from, int to); allows you to manipulate part of a list
A sublist may be used just like any other list

What is the java.util.Collections?

Offers many very useful utilities and algorithms for manipulating and creating collections Sortinglists, Index searching, Finding min/max, Reversing elements of a list, Swapping elements of a list, Replacing elements in a list and Other nifty tricks. Saves you having to implement them yourself and reuse

Simple java.util.Collections Code

```
package edu.batch.coll.list;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class CollectionsExample {
                    public static void main(String[] args) {
                         List<Integer> list = new ArrayList<Integer>();
                         list.add(78945);
                         list.add(85485);
                         list.add(12345);
                         list.add(11111);
                         list.add(22222);
                         list.add(58457);
                         list.add(11111);
                         System.out.println(list);
                         Collections.sort(list, Collections.reverseOrder());
                         System.out.println(list);
                         Collections.reverse(list);
                         System.out.println(list);
                         System.out.println(Collections.max(list));
                         System.out.println(Collections.min(list));
                         System.out.println(Collections.frequency(list, 11111));
                         Collections.replaceAll(list, 11111, 99999);
                         System.out.println(list);
                         Collections.swap(list, 0, list.size() - 1);
                         System.out.println(list);
                    }
}
```

What are two different ways of sorting collections?

Natural Ordering: Comparable method

External Ordering: Comparator method

Simple Collections.sort() Code: Students (Comparable: Natural Ordering)

```
package edu.batch.coll.sort;
import java.util.Date;
public class Student implements Comparable<Student> {
      private String name;
      private Byte age;
      private Integer studentId;
      private String course;
      private Date dob;

      @Override
      public String toString() {
            return "Student [name=" + name + ", age=" + age + ", studentId="
                        + studentId + ", course=" + course + ", dob=" + dob + "]";
      }
```

```
	public Student(String name, Byte age, Integer studentId, String course,
			Date dob) {
		super();
		this.name = name;
		this.age = age;
		this.studentId = studentId;
		this.course = course;
		this.dob = dob;
	}

	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}

	public Byte getAge() {
		return age;
	}

	public void setAge(Byte age) {
		this.age = age;
	}

	public Integer getStudentId() {
		return studentId;
	}

	public void setStudentId(Integer studentId) {
		this.studentId = studentId;
	}

	public String getCourse() {
		return course;
	}

	public void setCourse(String course) {
		this.course = course;
	}

	public Date getDob() {
		return dob;
	}

	public void setDob(Date dob) {
		this.dob = dob;
	}

	@Override
	public int compareTo(Student arg0) {
		return this.name.compareTo(arg0.name); // return 0 , 1, -1
	}
}
```

Simple Collections.sort() Code: Sorting Example

```
package edu.batch.coll.sort;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collections;
import java.util.Date;
import java.util.List;

public class SortingExample {
                    public static void main(String[] args) {
                         List<Student> students = new ArrayList<Student>();
                         students.add(new Student("John", (byte) 24, 12345,
"Computers",
                                   getDate(12, 04, 1991)));
                         students.add(new Student("Phil", (byte) 23, 85485,
"Electronics",
                                   getDate(05, 05, 1992)));
                         students.add(new Student("Carter", (byte) 25, 95848,
"Mathematics",
                                   getDate(1, 04, 1990)));
                         students.add(new Student("Stacy", (byte) 24, 96584,
"Computers",
                                   getDate(8, 12, 1991)));
                         students.add(new Student("Mary", (byte) 22, 23254,
"Electronics",
                                   getDate(6, 10, 1993)));
                         students.add(new Student("Brad", (byte) 23, 34792,
"Electronics",
                                   getDate(12, 5, 1992)));

                         Collections.sort(students, new CourseComparator());

                         for (Student student : students) {
                               System.out.println(student);
                         }

                    }

                    private static Date getDate(int month, int day, int year) {
                         Calendar c = Calendar.getInstance();
                         c.set(Calendar.DAY_OF_MONTH, day);
                         c.set(Calendar.MONTH, month);
                         c.set(Calendar.YEAR, year);
                         return c.getTime();
                    }
}
```

Simple Collections.sort() Code: Dob/Course (External Ordering: Comparator)

```
package edu.batch.coll.sort;

import java.util.Comparator;

public class DobComparator implements Comparator<Student> {
	@Override
	public int compare(Student arg0, Student arg1) {
		return arg1.getDob().compareTo(arg0.getDob());
	}
}
package edu.batch.coll.sort;

import java.util.Comparator;

public class CourseComparator implements Comparator<Student> {

	@Override
	public int compare(Student o1, Student o2) {
		return o1.getCourse().compareTo(o2.getCourse());
	}

}
```

What are the two important interfaces of Set?
Set and SortedSet are the two important interfaces of Set

What is the different between List and Queues?
Lists are not synchronized and queues are synchronized.

What is the difference between HashMap and Hashtable?
Hashmap can have null keys. Hashtable cannot have null keys. Hash Map is synchronized and Hashtable is not synchronized.

Which type of set is ordered?
LinkedHashSet is ordered.

Differentiate b/w HashSet/TreeSet?
HashSet: All operations are faster over here. It stores element in the form of bucket and it is divided into multiple chunks. Class offers constant time performance based on basic operations (add, remove, contains and size). It does not gurantee that the order of elements will remain constant over time. Iteration performance depends on the initial capacity and the load factor of the HashSet.
TreeSet: Gurantees log(n) time cost for the basic operations(add, remove and contains) Guarantees that elements of set will be sorted (Natural order). It doesn't offer any tuning parameters for iteration performance. It offers a few hand methods to deal with ordered set like first(), last().

Simple Set Code

```
package edu.batch.coll.set;

public class Employee implements Comparable<Employee> {
	private String name;
	private Integer id;

	@Override
	public String toString() {
		return "[name=" + name + ", id=" + id + "]";
	}

	public Employee(String name, Integer id) {
		super();
		this.name = name;
		this.id = id;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public Integer getId() {
		return id;
	}

	public void setId(Integer id) {
		this.id = id;
	}

	@Override
	public int compareTo(Employee o) {
		return this.name.compareTo(o.name);
	}

}
```

Simple Sorted Set Code

```
package edu.batch.coll.set;

import java.util.SortedSet;
import java.util.TreeSet;

public class SetExample {
	public static void main(String[] args) {
		Employee e1 = new Employee("Suraj", 987456);
		Employee e2 = new Employee("Sameer", 258471);
		Employee e3 = new Employee("Parul", 357958);
		Employee e4 = new Employee("Anirudh", 325847);
		Employee e5 = new Employee("Praveen", 842252);
		Employee e6 = new Employee("Akash", 965845);

		SortedSet<Employee> employees = new TreeSet<Employee>(
				new IdComparator());

		employees.add(e1);
		employees.add(e2);
		employees.add(e3);
		employees.add(e4);
		employees.add(e5);
		employees.add(e6);

		for (Employee employee : employees) {
			System.out.println(employee);
		}
		System.out.println("==================");
		System.out.println(employees.first());
		System.out.println(employees.last());

	}
}
```

Simple Sorted Set Code: Passing to the Comparator

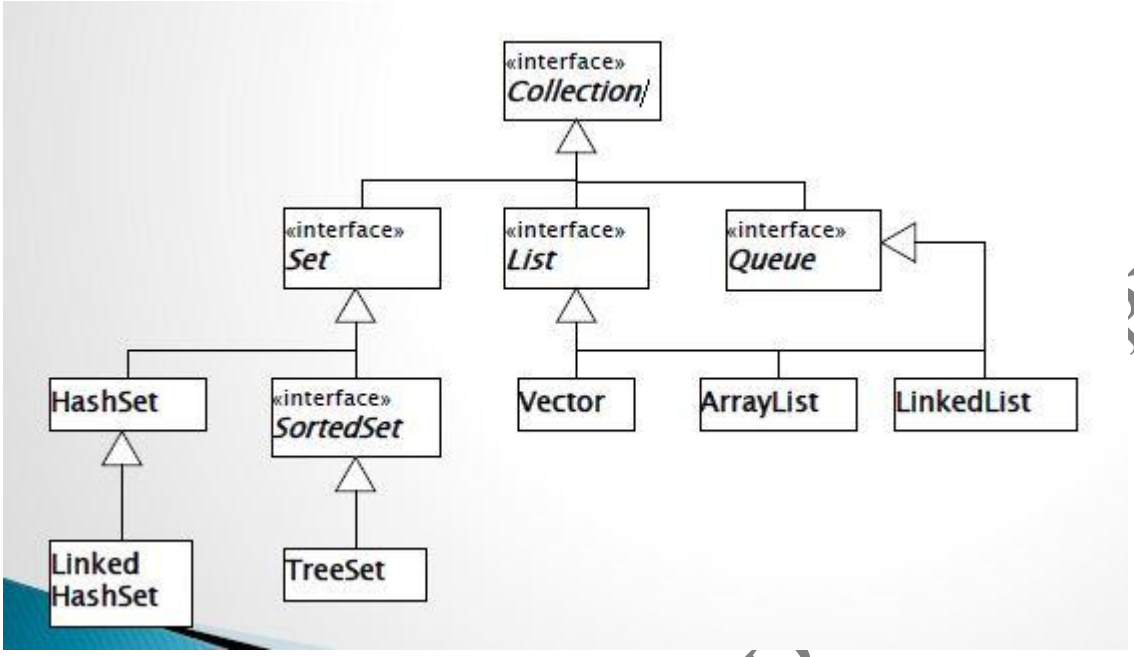
```
package edu.batch.coll.set;

import java.util.Comparator;

public class IdComparator implements Comparator<Employee> {

	@Override
	public int compare(Employee o1, Employee o2) {
		return o1.getId().compareTo(o2.getId());
	}
}
```

What is some of the feature of Map?
Map has the same structure as set. Map does not extend collections.

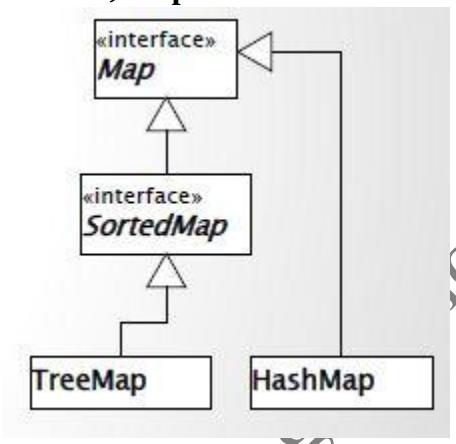
Simple Map Code

```
package edu.batch.coll.map;

import java.util.SortedMap;
import java.util.TreeMap;

public class MapExample {
                    public static void main(String[] args) {
                         SortedMap<String, Float> scores = new TreeMap<String,
Float>();
                         scores.put("CS", 3.5f);
                         scores.put("RT", 3.2f);
                         scores.put("EN", 3.2f);
                         scores.put("LS", 3.2f);
                         scores.put("CS", 4.5f);

                         System.out.println(scores);

                         System.out.println(scores.get("EN"));

                         System.out.println(scores.size());

                         scores.remove("LS");
                         System.out.println(scores);

                         System.out.println(scores.keySet());

                         System.out.println(scores.values());

                         System.out.println(scores.firstKey());

                         System.out.println(scores.lastKey());

                    }
}
```

In Brief, Collections



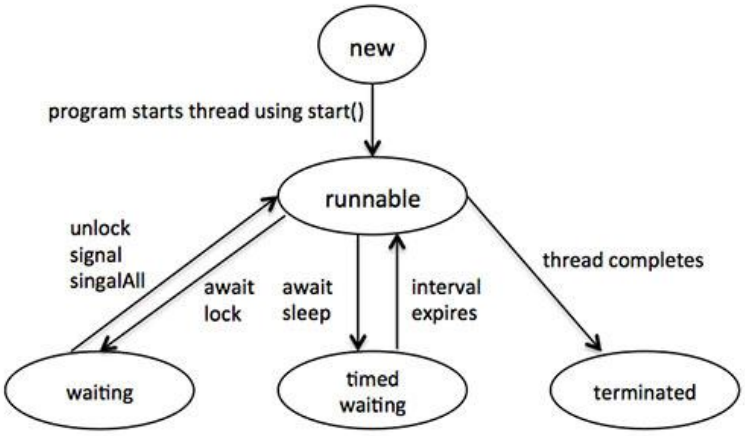
In Brief, Maps



CHAPTER: THREADING

What are the two different ways of creating threads?
By implementing runnable interface and extending the thread class.

What is the thread lifecycle?



At a particular instance only one process is occurring microsecond or millisecond processes.
Internally it is sequential, when the job is done, it is terminated.

What are the thread attributes?
It has priorities from 0 to 10 and its default value is 5
Every thread has its own memory (Stack size/ heap size) = Java thread size
Name can be named.
Thread group: Reading and writing to the database
Detach state
Scheduling Policy: Set manually or inherit from parent
Inherit Scheduling

What are some of the methods in threads?
public void start()
public void run()
public final void setName(String name)
public final void setPriority(intpriority)
public final void setDaemon(booleanon)
public final void join(long millisec)
public void interrupt()
public final booleanisAlive()
public static void yield()

Where we can use synchronize?
We can use synchronize method and synchronize block.

What are some of the thread operations?
public void suspend()
public void stop()
Public void resume()
public void wait()
public void notify()Wakes up a single thread that is waiting on this object's monitor.
public void notifyAll()Wakes up all the threads that called wait( ) on the same object.

What are some of the advantages and disadvantages of Threads?
Advantages

- Better Performance
- Share Resources
- Utilize Multiple Resources

Disadvantages

- Deadlock
- Switching Overhead

Simple Thread Code: OddThread using extends Thread

```
package edu.batch.thread;

public class OddThread extends Thread {
	public OddThread(ThreadGroup tg, String name) {
		super(tg, name);
	}

	public void run() {
		for (int i = 1; i <= 100; i++) {
			if (i % 2 == 1) {
				System.out.println(i);
				try {
					Thread.sleep(20);
				} catch (InterruptedException e) {
					e.printStackTrace();
				}
			}
		}
	}
}
```

Simple Thread Code: Thread Main

```
package edu.batch.thread;

public class ThreadMain {
	public static void main(String[] args) {
		ThreadGroup tg = new ThreadGroup("Numbers");
		OddThread ot = new OddThread(tg, "Odd"); // New State
		EvenThread et = new EvenThread(tg, "Even");

		ot.setPriority(Thread.MAX_PRIORITY);

		ot.setDaemon(true);

		ot.interrupt();
		ot.yield();

		ot.start();
	}
}
```

Simple Thread Code: Even Thread using implements Runnable

```
package edu.batch.thread;

public class EvenThread implements Runnable {
	public EvenThread(ThreadGroup tg, String name) {
		new Thread(tg, this, name).start();
	}

	public void run() {
		for (int i = 1; i <= 100; i++) {
			if (i % 2 == 0) {
				System.out.println(i);
				try {
					Thread.sleep(20);
				} catch (InterruptedException e) {
					e.printStackTrace();
				}
			}
		}
	}
}
```

Simple Thread Code: Explaining Synchronization (Account)

```
package edu.batch.thread;

public class Account {
	private String name;
	private Double balance;

	public Account(String name, Double balance) {
		this.name = name;
		this.balance = balance;
	}

	public synchronized void withdraw(Double reqAmt)
			throws InterruptedException {
		// fetch the balance
		Double tempBal = balance;
		Thread.sleep(100);

		// compare the amt with actual balance
		if (tempBal >= reqAmt) {
			tempBal = tempBal - reqAmt;
			System.out.println("Successfully withdrawn the amount");
		} else {
			System.out.println("Insufficient Funds");
		}

		this.balance = tempBal; // setting back the final balance

		System.out.println("The final balance is : " + this.balance);
	}
}
```

Simple Thread Code: Explaining Synchronization (ATM)

```
package edu.batch.thread;

public class ATM extends Thread {
	private Account acc;
	private Double reqAmt;

	public ATM(Account acc, Double reqAmt) {
		this.acc = acc;
		this.reqAmt = reqAmt;
	}

	public void run() {
		try {
			synchronized (acc) {
				acc.withdraw(reqAmt);
			}
		} catch (InterruptedException e) {
			e.printStackTrace();
		}
	}
}
```

Simple Thread Code: Explaining Synchronization (Sync Test)

```
package edu.batch.thread;

public class SyncTest {
	public static void main(String[] args) {
		Account acc = new Account("John/Shepard", 5000.00);

		ATM john = new ATM(acc, 4000.00);
		ATM shepard = new ATM(acc, 4000.00);

		john.start();
		shepard.start();

		john.suspend();
		john.resume();
		john.stop();

	}
}
```

Simple Thread Code: Explaining Thread Operations (HR)

```
package edu.batch.thread;

public class HR extends Thread {
	private String[] questions = { "What is you name?", "How old are you?",
			"What is your qualification?" };
	private Chat c;

	public HR(Chat c) {
		this.c = c;
	}

	public void run() {
		for (int i = 0; i < questions.length; i++) {
			try {
				c.ask(questions[i]);
			} catch (InterruptedException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		}
	}
}
```

Simple Thread Code: Explaining Thread Operations (Chat)

```
package edu.batch.thread;

public class Chat {
	boolean flag = true;

	public synchronized void ask(String q) throws InterruptedException {
		if (!flag) {
			wait();
		}
```

```
			System.out.println(q);
			flag = false;
			notify();
		}

		public synchronized void reply(String a) throws InterruptedException {
			if (flag) {
				wait();
			}
			System.out.println(a);
			flag = true;
			notify();
		}

}
```

Simple Thread Code: Explaining Thread Operations (Employee)

```
package edu.batch.thread;

public class Employee extends Thread {
	private String[] answers = { "Peter", "26", "Masters in Computers" };
	private Chat c;

	public Employee(Chat c) {
		this.c = c;
	}

	public void run() {
		for (int i = 0; i < answers.length; i++) {
			try {
				c.reply(answers[i]);
			} catch (InterruptedException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		}
	}
}
```

Simple Thread Code: Explaining Thread Operations (Wait Notify Main)

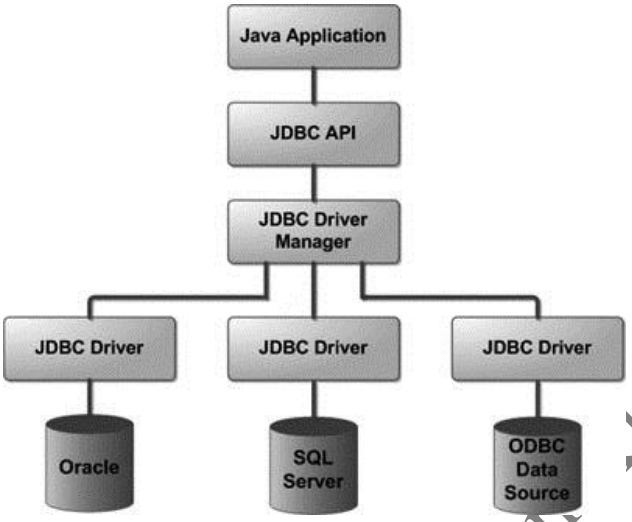
```
package edu.batch.thread;

public class WaitNotifyMain {
	public static void main(String[] args) {
		Chat c = new Chat();
		HR hr = new HR(c);
		Employee e = new Employee(c);

		hr.start();
		e.start();
	}
}
```

CHAPTER: JDBC

What is JDBC (Java Database Connectivity?

It can be organized in the form of table. The difference between excel and query is the level of access that is possible for both rows and columns. There is no concept of commit and roll back. The fast retrieval, packages, conditions in JDBC has a storage of some form. Java needs some kind of API, or built in program to connect with database. JDBC the queries, the viewing or modifying the database has better performance than ODBC. ODBC is the open database connectivity which has an open source framework.

What is the JDBC Architecture?



JDBC jar file will talk to the database with the information that the user wants.

What is the driver manager?

It manages list of database drivers and matches connection request with proper database driver.

Which component of JDBC directly communicates with the database?

The database driver

Using which object we perform all the operations in database?

Connection

What is DDL?

It is the data definition language. Ex: Boolean status=stmt.execute("ALTER TABLE patient")

What is DML?

It is the data manipulation language. Ex: int rowsChanged=stmt.execute Update("Update patient SET phone='xxx-xx-xxx')

What is DRL?
It is the data retrieval language. Ex: ResultSet res=stmt.executeQuery("SELECT p.name, p.disease, p.room.no);

What is Transaction?
Whenever there is transfer of data. It could be one program to another program or one location to another location.

What are the different types of statements in JDBC?
Simple Statements are static and not dynamic. It is used for general purposes access to database. It is used for executing static SQL. Statement interface cannot accept parameter. It has various methods
Prepared Statements: It extends statement interface. Flexibility of supplying arguments dynamically at runtime. Parameter are represent by "?" We bind the values using setxxx() methods
Callable Statement: It used to call database stored procedures. 3 types of parameters exists in, out and inout. We bind the In parameters using setXXX(). Bind the out parameters using registerOutParameters() and get the value using getXXX().

What are the driver types?
JDBC-ODBC bridge driver: It is the type 1 driver, JDBC call converts to ODBC driver then ODBC is responsible for database calls. This is extremely slow and backdated.

Native API: It is the type 2 driver, it converts Java calls to C/C++ programs then C/C++ is responsible for database calls. There is no overhead of ODBC.

Network Driver: It is the type 3 driver. It is also known as Middleware Driver. It has no API, one driver is responsible to choose any form. It happens through sockets.

100% Pure JDBC: It has no middleware drive, no bridge, and no conversion, at runtime we can download the driver dynamically.

What are the steps in building JDBC database?
First import Packages = java.sql.*classes
Register the JDBC driver – Class.forName()
Open the connection – DriverManager.getConnection()
Build the SQL statement- simple, prepared or callable.
Execute the query on the database- execute
Extract data from the Result.set-rs.getxxx()
Close the Result Set-rs.close()
Close the Statement-st.close()
Close the Connection-conn.close()
Handle exception if any – SQL exception

What are the various operations of a result set?
Rs.next(), Rs.next(), Rs.afterLast, beforeFirst, next, previous, absolute, relative are the various operations of a result set.

What are the properties of transaction?
Atomicity: All or nothing. If one part of the transaction fails, then the entire transaction fails.
Consistency: The logic and state of the transaction should be same as before and after
Isolation: No two transaction collides. Each transaction does know about other transactions. They are separate.
Durability: Ensures the once a transaction has been committed, it will remain so, even in the event of power loss, crashes or errors. It does not revert back to its old state.

What are the JDBC Batch Processing?
Group of related SQLs into a batch and execute them all once.

The methods of the JDBC Batch Processing?
The methods are addBatch(), executeBatch() and clearBatch()

What are the components of JDBC?
Driver Manager: manages list of DB Drivers; matches connection request with proper database driver
Driver: Handles communication with the database server
Connection: Interface for contacting a database; represents the communication context
Statement: For submitting SQL statements to the database
Result Set: Contains data retrieved from the database
SQL Exception

What are some of the syntax of SQL?
Create Database; Drop Database; Create Table; Drop Table; Insert Data; Select Data; Update Data; Delete Data.
What are some of the features of Transaction?
Commit
conn.setAutoCommit(bool)
conn.commit()
Rollback
conn.rollback()
Savepoint
setSavepoint(name)
releaseSavepoint(name)
conn.rollback(savepointName)

Simple JDBC Code: (Doctor)

```
package edu.batch.jdbc;

//Importing the necessary Classes
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DoctorDAO {

	public static void main(String[] args) {
		try {
			// Register the driver
			Class.forName("org.postgresql.Driver");
			/*
			 * Driver d = new Driver(); DriverManager.registerDriver(d);
			 */
			// Get the connection
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			// Create the Statement / Query
			Statement stmt = conn.createStatement();
			// Execute the Query
			ResultSet res = stmt
					.executeQuery("SELECT id, name, specialization,
location, phone FROM doctor");
			// Parse the result
			while (res.next()) {
				Integer id = res.getInt("id");
				String name = res.getString(2);
				String specialization = res.getString(3);
				String location = res.getString("location");
				String phone = res.getString("phone");

				System.out.println(id + "\t" + name + "\t" +
specialization
						+ "\t" + location + "\t" + phone);
			}

			// Close all
			res.close();
			stmt.close();
			conn.close();

			// Handling Exceptions
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}

	}
}
```

Simple JDBC Code: (Patient)

```
package edu.batch.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class PatientDAO {
	public static void main(String[] args) {
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			Statement stmt = conn.createStatement();

			// DDL - Data Definition Language
			boolean status = stmt
					.execute("ALTER TABLE patient ADD COLUMN phone
text");

			// DML - Data Manipulation Language
			int rowsChanged = stmt
					.executeUpdate("UPDATE patient SET phone = 'XXX-XX-
XXXX'");
			System.out.println("Rows updated  = " + rowsChanged);

			// DRL - Data Retrieval Language
			ResultSet res = stmt
					.executeQuery("SELECT p.name, p.disease, p.room_no,
d.name "
							+ "FROM patient p, doctor d where
p.doctor_id = d.id");
			while (res.next()) {
				System.out.println(res.getString(1) + "\t" +
res.getString(2)
						+ "\t" + res.getInt(3) + "\t" +
res.getString(4));
			}

			res.close();
			stmt.close();
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
	}
}
```

Simple JDBC Code: (Visitor)

```
package edu.batch.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class VisitorDAO {
	public static void main(String[] args) {
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			PreparedStatement pstmt = conn
					.prepareStatement("select id, name, phone from "
							+ "visitor where patient_id=?");
			pstmt.setInt(1, 102);
			ResultSet res = pstmt.executeQuery();
			while (res.next()) {
				System.out.println(res.getInt(1) + "\t" + res.getString(2)
						+ "\t" + res.getString(3));
			}
			System.out.println("================================");
			pstmt.setInt(1, 101);
			res = pstmt.executeQuery();
			while (res.next()) {
				System.out.println(res.getInt(1) + "\t" + res.getString(2)
						+ "\t" + res.getString(3));
			}

			res.close();
			pstmt.close();
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
	}
}
```

Simple JDBC Code: (Report)

```
package edu.batch.jdbc;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Types;

public class ReportDAO {
	public static void main(String[] args) {
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			CallableStatement cstmt = conn
					.prepareCall("{ CALL generate_report(?,?,?)}");
			boolean exists = false;
			int patientId = 102;

			cstmt.setInt(1, patientId);
			cstmt.setBoolean(2, exists);

			cstmt.registerOutParameter(2, Types.BOOLEAN);
			cstmt.registerOutParameter(3, Types.INTEGER);

			cstmt.execute();

			exists = cstmt.getBoolean(2);
			int visitorCount = cstmt.getInt(3);

			System.out.println(visitorCount);
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
	}
}
```

What is ResultSet Object? What are some of the features of it?
Any select query returns a result set object
Used for building the query response

Methods
beforeFirst–moves the cursor just before first row
afterLast–moves the cursor just after last row
first –moves the cursor to first row
last –moves the cursor to last row
absolute(rownum) –moves the cursor to rownum
relative(val) –moves the cursor valno.ofrows
previous –moves the cursor to previous row
next –moves the cursor to next row
getRow–returns the current row number

To retrieve a particular value we need to use the below methods:
rs.getXXX(colName)
rs.getXXX(colIndex)

Simple JDBC Code: Analysis (Result Set)

```
package edu.batch.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class AnalysisDAO {
	public static void main(String[] args) {
		try {
			// Register the driver
			Class.forName("org.postgresql.Driver");

			/*
			 * Driver d = new Driver(); DriverManager.registerDriver(d);
			 */

			// Get the connection
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");

			// Create the Statement / Query
			Statement stmt = conn.createStatement(
					ResultSet.TYPE_SCROLL_INSENSITIVE,
					ResultSet.CONCUR_READ_ONLY);
			ResultSet res = stmt.executeQuery("select d.name, count(d.name) "
					+ "from patient p, doctor d where p.doctor_id = d.id
"
```

```
                                    + "group by d.name order by d.name desc");

                        res.last();
                        System.out.println(res.getString(1) + "\t" + res.getInt(2));
                        System.out.println("====================================");
                        res.first();
                        System.out.println(res.getString(1) + "\t" + res.getInt(2));
                        System.out.println("====================================");
                        res.absolute(3);
                        System.out.println(res.getString(1) + "\t" + res.getInt(2));
                        System.out.println("====================================");
                        res.relative(2);
                        System.out.println(res.getString(1) + "\t" + res.getInt(2));
                        System.out.println("====================================");
                        res.previous();
                        System.out.println(res.getString(1) + "\t" + res.getInt(2) + "\t"
+ res.getRow());
                        System.out.println("====================================");

                        // Close all
                        res.close();
                        stmt.close();
                        conn.close();

                        // Handling Exceptions
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (ClassNotFoundException e) {
                        e.printStackTrace();
                  }

            }
}
```

Simple JDBC Code: Transaction

```
package edu.batch.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Savepoint;
import java.sql.Statement;

public class Transaction {
	public static void main(String[] args) {
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			conn.setAutoCommit(false);
			Statement stmt = conn.createStatement();
			Savepoint s = null;
			try {
				stmt.executeUpdate("UPDATE account SET balance=balance-
1000 WHERE name='A'");
				s = conn.setSavepoint();
				if (0 == 1) {
					throw new Exception("Some error");
				}
				stmt.executeUpdate("UPDATE account SET
balance=balance+1000 WHERE name='B'");
				conn.releaseSavepoint(s);
				conn.commit();
			} catch (Exception e) {
				e.printStackTrace();
				conn.rollback(s);
				//conn.rollback();
			}
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}

	}
}
```

Simple JDBC Code: Batch Processing

```
package edu.batch.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Date;
import java.util.Random;

public class Batch {
	public static void main(String[] args) {
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			PreparedStatement pstmt = conn
					.prepareStatement("INSERT INTO test_data(id, name,
amount) VALUES (?, ?, ?)");
			Random r = new Random();
			System.out.println(new Date());
			for (int i = 0; i < 10; i++) {
				pstmt.setInt(1, r.nextInt());
				pstmt.setString(2, "TEST - " + i);
				pstmt.setDouble(3, r.nextDouble());
				pstmt.addBatch();
			}
			pstmt.executeBatch();
			pstmt.clearBatch();
			System.out.println(new Date());
			pstmt.close();
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}

	}
}
```

CHAPTER: SERVLETS

What is servlet?
It is a server side program which allows you to execute a piece of code. It is a middle layer between the webpage and the server. In most of the cases, the starting point of any server will be servlet.

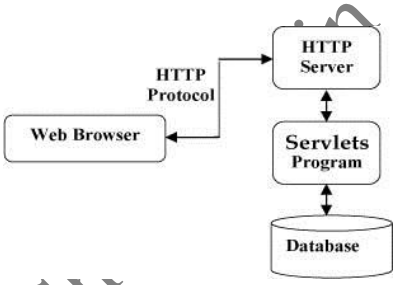
What is client?
A machine from which you are trying to send a request, send some orders, data or information. In web applications, browser becomes the client. In desktop application, the machine becomes the client.

What is Server?
It is a place where your application get hosted or deployed. It is the one which handles the request and repairs the response back to the client. On the server, everything is related to the application. Database server, application server are some of the examples of the server.
The communication happens through request and response. They are platform independent. Servlets are more secured. It can utilize the whole library.

What is a 3-tier Application?
Any client server application is called 2-tier architecture. The first tier is always the client (webpages), 2nd tier (java), 3rd tier (database).

What is CGI?
It is known as common gateway interface. It is an old technology which is replaced by servlets. Also, the implementation wise it wasn't efficient.

What is the Servlet Architecture?

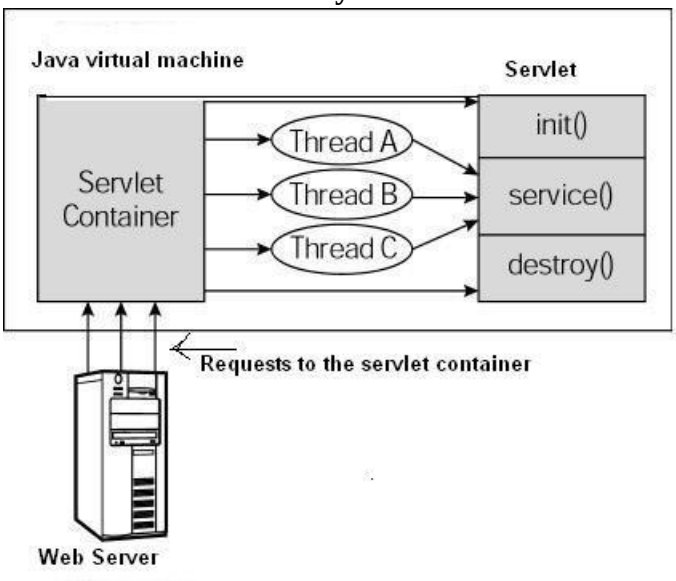


Read implicit & explicit data from clients
Sends implicit & explicit data to the clients
Process data and generate results

What is the implicit data?

Implicit data: It is such data which is not required to manually enter. Hence, it automatically goes. They are by default send to the server. Ex: Google cookies remembering the user id and the password. Therefore, the cookies automatically it goes to the server. It remembers the IP address, date and time, which location you are trying to hit from.

What is the servlets lifecycle?



Init(): It stands for initialization. To perform some logic, it needs to state in this method. There are two type of Init() methods. One is without parameters and the other is with servlet config parameter. All start up activity and initialization activity. Init() method get invoked in the servlet only for the first time. It is always more client one server.

Service(): It is where you write down the business logic. It takes two parameters: servlet request and servlet response. It gets called on each hit. Ex: Transfer money, register.

Destroy(): It does not take any parameters and does not return anything. When we want to undeploy from the server.

What is the Load on start up?

Load on startup flag which loads the servlets automatically, before the requests gets hit.

What are the types of servlet?

Generic Servlet: It is the superclass of all servlets. It is used to implement client server interaction and it is used for non-browser application. Ex: Email application, Skype, G talk, File transfer protocol. Service () method is used for execution. All Init(), Service() and Destroy used.

Hypertext transfer protocol: A child of generic servlet is HTTP servlet. It is the most popular servlet in the market today. It is used for web application. We used two methods doGet (Getting some information from the server and doPost (Posting from the server).

Simple Code Servlet: Login Servlet

```
package com.servlet;

import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

import javax.servlet.ServletConfig;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {

	public void init(ServletConfig config) {
		String message = config.getInitParameter("message");
		System.out.println(message);
	}

	public void doGet(HttpServletRequest request, HttpServletResponse response)
			throws IOException {
		String username = request.getParameter("username");
		String password = request.getParameter("password");

		Cookie c = new Cookie("creds", username + "|" + password);
		response.addCookie(c);

		if ("admin".equals(username) && "test321".equals(password)) {
			response.getWriter().write("Successful Login");
		} else {
			response.getWriter().write("Login failed");
		}
		response.getWriter().write(request.getQueryString());

		response.getWriter().write(request.getRequestURI());

		BufferedInputStream bis = new BufferedInputStream(
				request.getInputStream());
		FileOutputStream fis = new FileOutputStream(new File(
				"c:\\temp\\stream.txt"));
		byte[] bytes = new byte[1000];
		bis.read(bytes);
		fis.write(bytes);
		fis.close();
		//response.setContentType("application/txt");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
			throws IOException {
```

```
		System.out.println("I am in post");
		doGet(request, response);
	}

	public void destroy() {
		System.out.println("Destroying the servlet.");
	}
}
```

Simple Code Servlet: Web XML

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns="http://java.sun.com/xml/ns/javaee"
	xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
	id="WebApp_ID" version="3.0">
	<display-name>Batch77WebProject</display-name>
	<servlet>
		<servlet-name>login</servlet-name>
		<servlet-class>com.servlet.LoginServlet</servlet-class>
		<init-param>
			<param-name>message</param-name>
			<param-value>This is my first servlet.</param-value>
		</init-param>
	</servlet>
	<servlet-mapping>
		<servlet-name>login</servlet-name>
		<url-pattern>/login.chk</url-pattern>
	</servlet-mapping>
	<welcome-file-list>
		<welcome-file>login.html</welcome-file>
	</welcome-file-list>
</web-app>
```

Simple Code Servlet: Login HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<form action="Login.chk" method="get">
		<div>
			Username : <input type="text" name="username" id="username" />
		</div>
		<div>
			Password : <input type="password" name="password" id="password"
/>
		</div>
		<div>
			<input type="submit" value="Login" />
		</div>
	</form>
</body>
</html>
```

What are the HTTP Servlet Request Methods?

It is request object that comes from the client to the server. Implicit data or the explicit data will be built in the request object.

- getCookies: All the cookies responding to the website, which will return the available cookies and perform operations.
- getSession: It is one communication channel between the client and the servers. From the time your first request goes to your last request.
- getAttribute: Attribute name and the value are passed. Life of attributes till it exists.
- getParameterNames: Part of the request URL, will return only the username and password.
- getInputStream: Read the input in the form of the stream. Serialize it! Binary form
- getParameter: Will get the value for particular parameter.

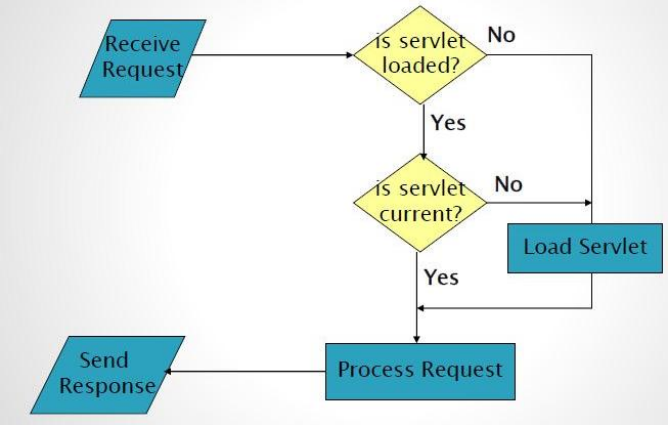
//localhost8080/Batch77WebProject/login.ck?username=admin&password=test321

- getRequestURI: It is the old path of the url, ***login.ck?username=admin&password=test321***
- getServletPath: ***Batch77WebProject/login.ck?***
- getQueryString: ***username=admin&password=test321***

What are the HTTP Servlet Response Methods?

addCookie:
sendRedirect: Redirect the page to some other page based on the request
setHeader: Any information which requires on the title. Ex: Encoding to be used, content
flushBuffer: Flash the response on the screen, you cannot modify the response
reset: Clean all the response, go back to the initial state.
getWriter: It's a handle to write date/output to the browser
setContentType: It is where we can have the option for: whether the output should be html or not
setBufferSize: Limit the size of the response

What is the working of a servlet?



Once the request is receiver, servlet has been initialized or not, is the servlet that I require then it will call the do get and do post method accordingly.

What is the difference between the servlets and java application?

Java application has a main method but servlet do not have any main functions. Starting point is either doGet or doPost method. Invocation happens via the browser. Interaction via request/response object APIs. In case of main method you can have parameters but you cannot return anything. Primary servlet output is typically HTML. Servlet are sitting inside a Servlet Container (Server).

What are the parameters for each of the method?
Init method takes servlets config
Service methods: doGet or doPost, request and response
Destroy doesn't take any parameters

What is the difference between doGet and doPost method?
DoGet: It is used to get the parameters. doPost is used to post any parameters to the server. For doGet method the parameters are displayed in the url but for the doPost method the parameters are not displayed in the url.

How do you configure servlets? How do connect the servlet from the front end to the back end?
By using web.xml. First define the servlet name and class. Using servlet mapping which contains servlet name and url pattern.

What are the two ways of validating the XML? What are the difference between them?
Using DTD or XSD. A DTD describes the tree structure of a document and something about its data. There are two data types, PCDATA and CDATA. PCDATA is parsed character data where there is no ampersand, CDATA is character data, not usually parsed. A DTD determines how many times a node may appear, and how child nodes are ordered. In Schemas we have more complex types. Schemas have more data types, DTD are not as comprehensive as Schemas. Schemas divide the tasks into simple and complex. The complex has children but the simple doesn't have any children.

Can you define repetitions in DTDS?
Yes, we can specifically say. * which represents 0 or more. < for one or more. ? for may or may not occur 0 or 1. We can do use min occurrence or maximum occurrences in schemas.

Chapter: XML Programming

What is XML?

XML stands for Extensible Markup Language.
It provide information about a document.
Tags are added to the document to provide the extra information.
HTML tags tell a browser how to display the document.
XML tags give a reader some idea what some of the data means.

Simple XML Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
	<BANKTRANSFER>
		<transactionNo id="123456789" />
		<fromPhone>732-854-9854</fromPhone>
		<toPhone>765-985-7854</toPhone>
		<fromBirth>1995-10-15</fromBirth>
		<toBirth>1995-02-02</toBirth>
		<fromName>Adam &lt; Sandler</fromName>
		<toName>Sandra &amp; Bullock</toName>
		<source />
	</BANKTRANSFER>
	<second>
	</second>
</root>
```

Where is XML used?

XML documents are used to transfer data from one place to another often over the Internet.
XML subsets are designed for particular applications. One is RSS (Rich Site Summary or Really Simple Syndication). It is used to send breaking news bulletins from one web site to another.
A number of fields have their own subsets. These include chemistry, mathematics, and books publishing. Most of these subsets are registered with the W3Consortium and are available for anyone's use.

What are the advantages of XML?

- XML is text (Unicode) based.
- Takes up less space.
- Can be transmitted efficiently.
- One XML document can be displayed differently in different media.
- Html, video, CD, DVD,
- You only have to change the XML document in order to change all the rest.
- XML documents can be modularized. Parts can be reused.

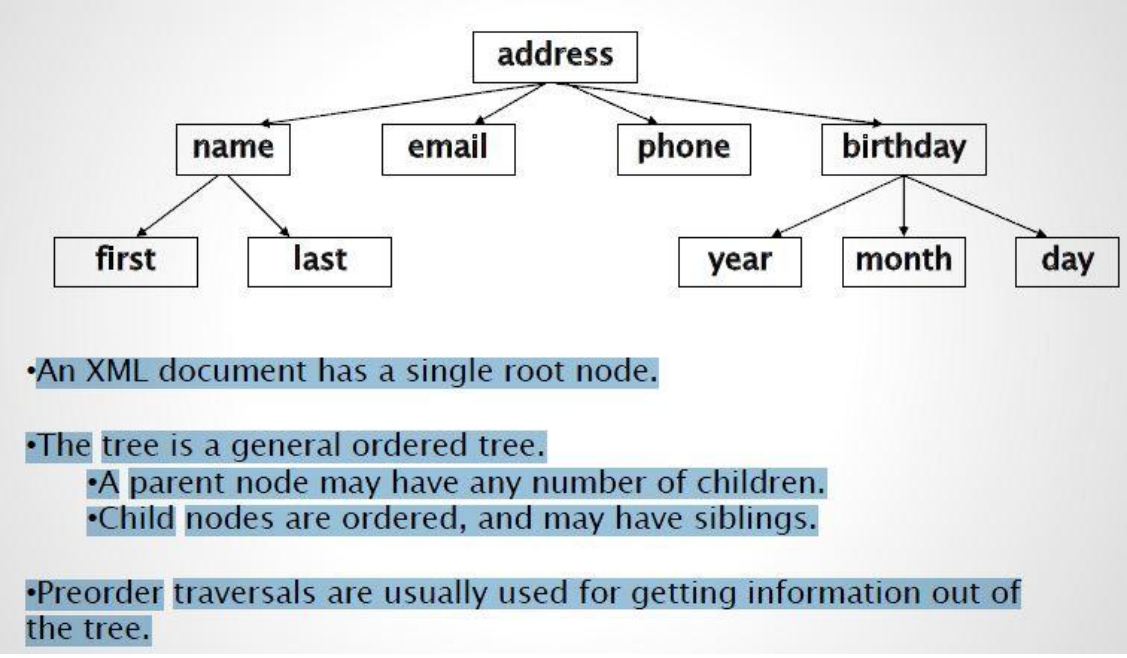
What are the differences between HTML and XML?

- HTML tags have a fixed meaning and browsers know what it is.
- XML tags are different for different applications, and users know what they mean.
- HTML tags are used for display.
- XML tags are used to describe documents and data.

What are the XML Rules?

- Tags are enclosed in angle brackets.
- Tags come in pairs with start-tags and end-tags.
- Tags must be properly nested.
- <name><email>…</name></email> is not allowed.
- <name><email>…</email><name> is.
- Tags that do not have end-tags must be terminated by a '/'.
-
 is an html example.
- Tags are case sensitive.
- <address> is not the same as <Address>
- XML in any combination of cases is not allowed as part of a tag.
- Tags may not contain '<' or '&'.
- Tags follow Java naming conventions, except that a single colon and other characters are allowed. They must begin with a letter and may not contain white space.
- Documents must have a single root tag that begins the document.

How is XML structured?



What is Validity?
A well-formed document has a tree structure and obeys all the XML rules. A particular application may add more rules in either a DTD (document type definition) or in a schema. Many specialized DTDs and schemas have been created to describe particular areas. These range from disseminating news bulletins (RSS) to chemical formulas. DTDs were developed first, so they are not as comprehensive as schema

What are the two types of XML?
You can add more rules with XML. DTD or Schemas (XSD). Even HTML have DTDs. They all follows the standards.

What are Document Type Definitions (DTD)?
A DTD describes the tree structure of a document and something about its data.
There are two data types, PCDATA and CDATA. PCDATA is parsed character data.
CDATA is character data, not usually parsed. A DTD determines how many times a node may appear, and how child nodes are ordered. This is simple

What are schemas?
Schemas are themselves XML documents. They were standardized after DTDs and provide more information about the document. They have a number of data types including string, decimal, integer, boolean, date, and time. They divide elements into simple and complex types. They also determine the tree structure and how many children a node may have. It has more features regarding to DTD.

Simple XML: DTD Code: Personal

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE client PUBLIC "" "client.dtd">
<client id="12345">
	<personal>
		<name>Peter Hood</name>
		<address>
			<street>#1 Park Ave</street>
			<city>Santa Clara</city>
			<state>CA</state>
			<zip>95439</zip>
		</address>
		<dob>1974-04-20</dob>
		<phone>874-987-8548</phone>
		<phone>879-854-7854</phone>
		<email>peter@hotmail.com</email>
		<blog>blogspot.com/peter</blog>
	</personal>
</client>

<!--
	? - optional (0 or 1)
	* - 0 or more
	+ - 1 or more
 -->
```

Simple XML: DTD Code: Client

```
<!ELEMENT client (personal)>
<!ATTLIST client
	id CDATA #REQUIRED
>
<!ELEMENT personal (name,address,dob,phone+,email,blog?)>
<!ELEMENT address (street,city,state,zip)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
<!ELEMENT dob (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT blog (#PCDATA)>
```

Simple XML: DTD Code: Account

```
<?xml version="1.0" encoding="UTF-8"?>
<account xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:noNamespaceSchemaLocation="resources/account.xsd">
	<name>Karl Marx</name>
	<accNo>8885222155</accNo>
	<routingNo>95184627</routingNo>
	<type>checking</type>
	<bank>Bank of America</bank>
	<branch>Lexington Ave</branch>
	<cardNo>7894-6512-3256-8547</cardNo>
	<ssn>789-54-6254</ssn>
	<phone>852-741-9632</phone>
	<email>karl.marx@microsoft.com</email>
	<address>
		<street>#1 Lexington Ave</street>
		<city>New York</city>
		<state>NY</state>
		<zip>65847</zip>
	</address>
	<startDate>2011-02-25</startDate>
</account>
```

Simple XML: XSD Code: Account

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
	<xs:element name="account">
		<xs:complexType>
			<xs:sequence>
				<xs:element type="xs:string" name="name" />
				<xs:element type="xs:long" name="accNo" />
				<xs:element type="xs:long" name="routingNo" />
				<xs:element type="xs:string" name="type" />
				<xs:element type="xs:string" name="bank" />
				<xs:element type="xs:string" name="branch" />
				<xs:element type="xs:string" name="cardNo" />
				<xs:element type="xs:string" name="ssn" />
				<xs:element type="xs:string" name="phone" minOccurs="1"
					maxOccurs="2" />
				<xs:element type="xs:string" name="email" />
				<xs:element name="address">
					<xs:complexType>
						<xs:sequence>
							<xs:element type="xs:string"
name="street" />
							<xs:element type="xs:string"
name="city" />
							<xs:element type="xs:string"
name="state" />
							<xs:element type="xs:int" name="zip" />
						</xs:sequence>
					</xs:complexType>
				</xs:element>
				<xs:element type="xs:date" name="startDate" minOccurs="0"
/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
</xs:schema>
```

Simple XML: XSD Code: Test Class

```
package com.xml;

import java.io.File;
import java.io.IOException;

import javax.xml.XMLConstants;
import javax.xml.transform.stream.StreamSource;
import javax.xml.validation.Schema;
import javax.xml.validation.SchemaFactory;
import javax.xml.validation.Validator;

import org.xml.sax.SAXException;

public class XSDTest {

	public static void main(String[] args) {

		try {
			SchemaFactory factory = SchemaFactory
					.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI);
			Schema schema = factory
					.newSchema(new File(

	"C:\\CMASWorkpace\\Batch77WebProject\\resources\\account.xsd"));
			Validator validator = schema.newValidator();
			validator
					.validate(new StreamSource(
							new File(

	"C:\\CMASWorkpace\\Batch77WebProject\\resources\\account.xml")));
			System.out.println("XML Validation Successful");
		} catch (IOException | SAXException e) {
			System.out.println("Exception: " + e.getMessage());
		}

	}
}
```

What is Parsing?
Reading the XML document, fetching the data inside the document is called Parsing and the storing the data in the database. In the ground level, JDK provides two kinds of parsing one is SaxParser or DomParser.

What are the features of Sax Parser?

- SAX –Simple API for XML
- Reading the XML document through a document.
- SAX –Simple API for XMLs
- Uses a call-back method
- Similar to javaxlisteners
- Parses node by node
- Doesn't store the XML in memory
- We can't insert or delete a node
- SAX is an event based parser
- SAX is a Simple API for XML
- doesn't preserve comments
- SAX generally runs a little faster than DOM

Simple Sax Parser Code:

```
package com.xml;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.SAXException;

public class AccountSAXParser {
	public static void main(String[] args) throws ParserConfigurationException,
			SAXException, IOException {
		SAXParserFactory factory = SAXParserFactory.newInstance();
		SAXParser parser = factory.newSAXParser();

		File input = new File(

	"C:\\CMASWorkpace\\Batch77WebProject\\resources\\account.xml");
		AccountHandler ah = new AccountHandler();
		parser.parse(input, ah);
	}
}
```

Simple Sax Parser Code: Account Handler

```
package com.xml;
import org.xml.sax.Attributes;
import org.xml.sax.helpers.DefaultHandler;

public class AccountHandler extends DefaultHandler {
	@Override
	public void startDocument() {
		System.out.println("Parsing started");
	}

	@Override
	public void startElement(String tag, String q, String qName,
			Attributes attrs) {
		System.out.println("START" + qName);
	}

	@Override
	public void endElement(String tag, String q, String qName) {
		System.out.println("END" + qName);
	}

	@Override
	public void characters(char[] text, int start, int length) {
		String body = new String(text).substring(start, start + length);
		if (body != null && body.trim().length() > 0) {
			System.out.println(body);
		}
	}

	@Override
	public void endDocument() {
		System.out.println("End of Parsing");
	}
}
```

What are the features of Dom Parser?

- DOM –Document Object Model
- Creates a parse tree
- Requires a tree traversal
- Stores the entire XML document into memory before processing
- Occupies more memory
- We can insert or delete nodes
- Traverse in any direction.
- DOM is a tree model parser
- Document Object Model (DOM) API
- Preserves comments
- DOM generally runs a little slower than SAX

Simple Dom Parser Code:

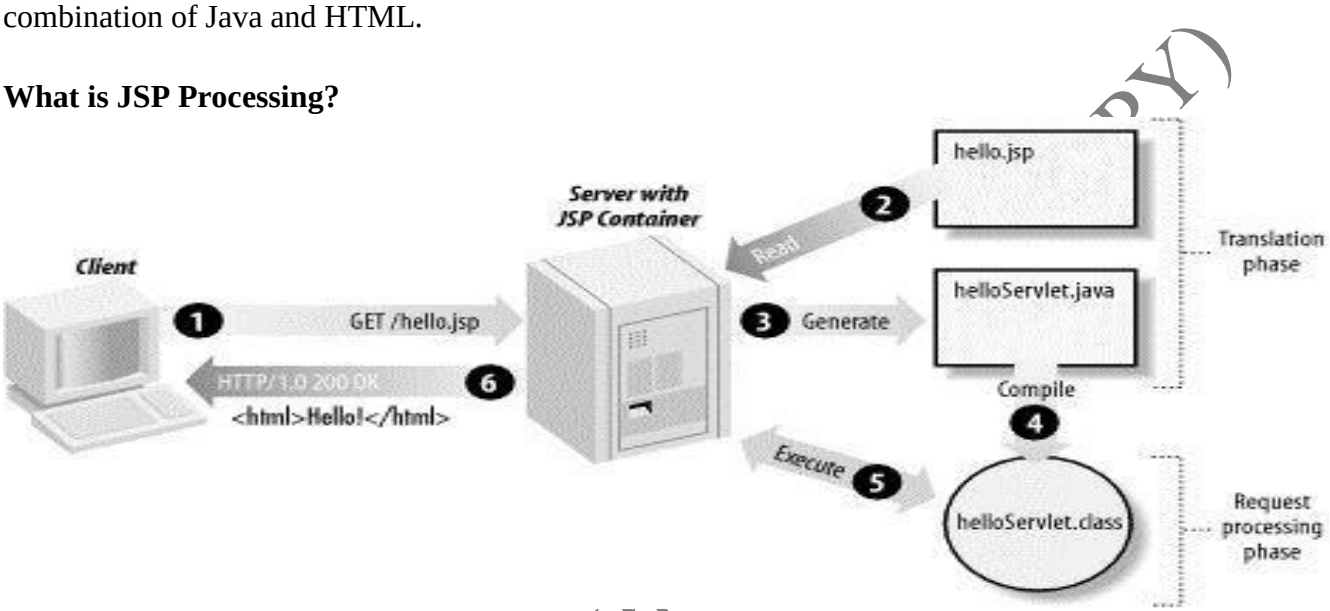
```
package com.xml;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class ClientDomParser {
	public static void main(String[] args) throws ParserConfigurationException,
			SAXException, IOException {
		DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
		DocumentBuilder parser = factory.newDocumentBuilder();

		Document doc = parser
				.parse(new File(

	"C:\\CMASWorkpace\\Batch77WebProject\\resources\\individual.xml"));
		NodeList nodes = doc.getElementsByTagName("phone");
		for (int i = 0; i < nodes.getLength(); i++) {
			Node node = nodes.item(i);
			System.out.println(node.getTextContent());
		}
		System.out.println(doc.getDocumentElement().getTagName());
	}
}
```

CHAPTER: JSP

What is JSP? What are the features of JSP?

It is the java server pages. JSP are used to create webpages for java based applications. Display the response coming from the server. First it is a java based component. Every JSP is internally a servlet. They are residing on a server. They are managed by the servlet container. JSP is the combination of Java and HTML.

What is JSP Processing?



Example of HTML

```
<html>
  <head><title>Example</title></head.
<body>
  <h1>This is an example of a page.</h1>
  <h2>Some information goes here.</h2>
</body>
</html>
```

Example of XML

```
<?xml version="1.0"/>
<address>
  <name>Alice Lee</name>
  <email>alee@aol.com</email>
  <phone>212-346-1234</phone>
  <birthday>1985-03-22</birthday>
</address>
```

Example of JSP

Scriptlet

```
<% code fragment %>, <%! declaration; [ declaration; ]+ ... %>, <%= expression
  %>
```

Simple JSP

```
<html>
   <head>
         <title>Hello World</title>
   </head>
   <body>
         Hello World!<br/>
         <% out.println("Your IP address is " + request.getRemoteAddr()); %>
   </body>
</html>
```

Simple JSP Code: Scriplet

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <!-- DECLARATION SCRIPTLET -->
      <%!String message = "Welcome to Java Server Pages.";%>

      <!-- CODE FRAGMENT -->
      <%
            int a = 44, b = 92;
            if (a < 50) {
                  a = a - 10;
            }
            if (b > 100) {
                  b = b - 20;
            }
      %>

      <!-- EXPRESSION SCRIPTLET -->
      <%=message%>
      <%=a + b%>
```

```
	<!-- CODE AND EXPRESSION -->
	<%
		for (int x = 1; x <= 10; x++) {
	%>
			<%=x%>
	<%
		}
	%>

</body>
</html>
```

What are some JSP Directives?

<%@ page ... %>

Defines page-dependent attributes, such as scripting language, error page, and buffering requirements.
Attributes:contentType, errorPage, isErrorPage, extend, Import, session, isScriptingEnabled

<%@ include ... %>

Includes a file during the translation phase.
Attributes:file

<%@ taglib... %>

Declares a tag library, containing custom actions, used in the page
Attributes:uri, prefix
<%@tagliburi="http://www.springframework.org/tags/form" prefix="form"%>

Simple JSP Code: Directives

```
<%@ page import="java.util.Date"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<%@ page errorPage="error.jsp" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <%
            int length;
            try {
                  String s = null;
                  length = s.length();
            } catch (Exception e) {
                  throw e;
            }
      %>
      <%="The length is : " + length%>
      <%=new Date()%>
      <%=session.getId()%>
</body>
</html>
```

Simple JSP Code: Error

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<%@ page isErrorPage="true"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      The system is currently facing issues. Please login after some time.

      <%=exception.getMessage()%>
</body>
</html>
```

Simple JSP Code: Include

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <h1>MAIN HEADER</h1>

      <h3>INCLUDED CONTENT</h3>
      <%@ include file="page.jsp"%>
</body>
</html>
```

Simple JSP Code: Tag library

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      The amount is : <c:out value="$20,000.00" />
</body>
</html>
```

What are the JSP actions?

JSP include (page): It is to include the page from one page to another. It is more dynamic. Only .jsp page can be included.
JSP useBean(id, class):
JSP setProperty(name, property, value):
JSP getProperty(name, property):
JSP forward (page): Forward request to some other page

Simple JSP Action Code: Forward

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<jsp:forward page="action.jsp">
		<jsp:param value="Java Server Pages" name="technology" />
	</jsp:forward>
</body>
</html>
```

Simple JSP Action Code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<%=request.getParameter("technology")%>

	<h3>SERVLET RESPONSE</h3>
	<jsp:include page="/login.chk?username=admin&password=test321"></jsp:include>
</body>
</html>
```

Simple JSP Action Code: Bean

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <jsp:useBean id="product" class="com.jsp.bean.Product" scope="request">
      </jsp:useBean>
      <jsp:setProperty property="name" name="product" value="Reebok Shoes" />
      <jsp:setProperty property="price" name="product" value="119.99" />
      <jsp:setProperty property="quantity" name="product" value="2" />

      <table>
            <tr>
                  <td>Product Name</td>
                  <td><jsp:getProperty property="name" name="product" />
            </tr>
            <tr>
                  <td>Product Price</td>
                  <td><jsp:getProperty property="price" name="product" />
            </tr>
            <tr>
                  <td>Product Quantity</td>
                  <td><jsp:getProperty property="quantity" name="product" />
            </tr>
      </table>

</body>
</html>
```

What are the Implicit Objects?

These object are by default available in the JSP.
Request (HTTPServletRequest): For to get and to post
Response (HTTPServletResponse): For to get and to post
Out (PrintWriter): Response. Get writer, writer is coming out as out object
Session (HTTPSession):
Application (ServletContext):
Config (ServletConfig):
Page (this JSP): Specifically for JSP attributes that can accessed within that page
Exception (Exception Object):

Simple JSP Implicit Code:

```
<%@page import="java.util.Date"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <div><%=request.getRequestURI()%></div>
      <div><%=response.getContentType()%></div>
      <div>
            <%
                  out.println("This is a message from out object.");
            %>
      </div>
      <div><%=new Date(session.getCreationTime())%></div>
      <div><%=application.getContextPath()%></div>
      <div><%=config.getServletName()%></div>
      <div><%=page.hashCode()%></div>
</body>
</html>
```

What are the HTTP Status codes?

The status codes are responsible for any back end calls. Whenever we make a back in call to the server from the client, the client not only response, it also gives back the status code.

200 –The request is OK
302 –Found (moved to a new URL temporarily)
400 –Bad Request
401 –Unauthorized (username, password)
403 –Forbidden
404 –Not Found
408 –Request Timed out
500 –Internal Server Error
503 –Service Unavailable

What are the JSTL (Java Serve Tag library)?

It is provide us with some predefined tags provided by the JSTL jar. JSTL is an external library. It comes with 5 tag libraries. They are XQL, XML, Core, Formatting and JSTL. XQL tags are used to connect to the database, get the data from the database to all the operators of the database. XML tags are used for performing XML transformation or XML parsing.

Core Tags

Out, Set, If, Choose, When, Otherwise, forEach, Param, Redirect, Url

Formatting Tags

formatDate, parseDate, message, formatNumber, parseNumber

JSTL Functions

Contains, endsWith, indexOf, join, length, replace, split, trim

Tags are used to overcome the problem of scriplets.

Simple JSTL code: Core

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<c:set var="language" value="English"></c:set>
	<c:out value="${language}" />
	<c:remove var="language" />

	<c:set var="bit" value="0"></c:set>
	<c:if test="${bit == 0 }">
		<div>The bit value is not set</div>
	</c:if>

	<c:set var="score" value="72" />
	<div>
		<c:choose>
			<c:when test="${score < 35 }">
				FAILED
			</c:when>
			<c:when test="${score < 70 }">
				PASSED
			</c:when>
			<c:when test="${score >= 70 && score <= 100 }">
				DISTINCTION
			</c:when>
			<c:otherwise>
				INVALID SCORE
			</c:otherwise>
		</c:choose>
	</div>
```

```
	<c:forEach var="i" begin="5" end="100" step="5">
		<div>${i }</div>
	</c:forEach>

	<c:url var="login"
		value="http://localhost:8080/Batch77WebProject/login.chk"></c:url>
	<c:redirect url="${login }">
		<c:param name="username" value="admin"></c:param>
		<c:param name="password" value="test321"></c:param>
	</c:redirect>

</body>
</html>
```

Simple JSTL code: Format

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<c:set var="dobStr" value="28/10/1990 17:45" />
	<fmt:parseDate pattern="dd/MM/yyyy HH:mm" value="${dobStr }"
var="dob"></fmt:parseDate>
	<fmt:formatDate value="${dob }" pattern="MM-dd-yyyy hh:mm a" />

	<c:set var="incomeStr" value="2510850" />
	<fmt:parseNumber value="${incomeStr }" integerOnly="true"
var="income"></fmt:parseNumber>
	<fmt:formatNumber currencySymbol="$" type="currency"
		groupingUsed="true" minFractionDigits="2" value="${income
}"></fmt:formatNumber>
	<fmt:setBundle basename="com.jsp.Messages" var="myMsgs" />
	<fmt:message key="welcome" bundle="${myMsgs }"></fmt:message>
	<fmt:message key="content" bundle="${myMsgs }"></fmt:message>
	<fmt:message key="conclusion" bundle="${myMsgs }"></fmt:message>
</body>
</html>
```

Simple JSTL code: JSP/ Messages Java

```
package com.jsp;

import java.util.ListResourceBundle;

public class Messages extends ListResourceBundle {

	@Override
	protected Object[][] getContents() {
		return new Object[][] { { "welcome", "Welcome to JSP Programming" },
					{ "content", "This is a class 3 on JSP" },
					{ "conclusion", "Thank you and come again" } };
	}
}
```

Simple JSTL code: Functions

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<c:set var="msg" value="The quick brown fox jumps over a lazy dog." />

	<c:if test="${fn:contains(msg, 'fox') }">
		This message is about fox
	</c:if>
	<div>Index of lazy is ${fn:indexOf(msg, 'lazy') }</div>

	<c:set var="words" value="${fn:split(msg,' ') }" />
	<c:forEach items="${words }" var="word">
		<div>${word }</div>
	</c:forEach>
	<c:set var="msg" value="${fn:join(words, ',') }" />

	<c:out value="${fn:toLowerCase(msg) }" />

	<c:out value="${fn:toUpperCase(msg) }" />
</body>
</html>
```

Simple JSTL code: Simple Tag Support (JSP)

```
package com.jsp;
import java.io.IOException;
import javax.servlet.jsp.tagext.SimpleTagSupport;

public class PhoneTag extends SimpleTagSupport {
	private String text;
	private final String numExpr = "^[0-9]+$";

	public void setText(String text) {
		this.text = text;
	}

	public void doTag() throws IOException {
		if (text != null && text.length() == 10 && text.matches(numExpr)) {
			String stCode = text.substring(0, 3);
			String lclCode = text.substring(3, 6);
			String last4 = text.substring(6);

			String result = "(" + stCode + ")" + lclCode + "-" + last4;
			getJspContext().getOut().write(
					"<font color='green'>" + result + "</font>");
		} else {
			getJspContext().getOut().write(
					"<font color='red'>" + text + "</font>");
		}
	}
}
```

Simple JSTL code: Web-INF App

```
<?xml version="1.0" encoding="UTF-8" ?>

<taglib xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
	version="2.1">

	<display-name>App Tags</display-name>
	<tlib-version>1.1</tlib-version>
	<short-name>app</short-name>

	<tag>
		<name>phone</name>
		<tag-class>com.jsp.PhoneTag</tag-class>
		<body-content>scriptless</body-content>
		<attribute>
			<name>text</name>
			<required>true</required>
		</attribute>
	</tag>
</taglib>
```

Simple JSTL code: Custom JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="app" uri="/WEB-INF/app.tld"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <app:phone text="7326585165"></app:phone>

      <app:phone text="9518462"></app:phone>
</body>
</html>
```

What are the steps for Custom Tag?

Steps involved in writing a Custom Tag

Create a Tag class by extending SimpleTagSupportand implement the doTagmethod

Create a custom tldfile

<taglib>
<tlib-version>1.0</tlib-version>
<jsp-version>2.0</jsp-version>
<short-name>CustomTag</short-name>
<tag>
<name>Test</name>
<tag-class>com.proj.TestTag</tag-class>
<body-content>empty</body-content>
</tag>
</taglib>

Use the tag in the jsp

<%@taglibprefix="t"uri="WEB-INF/custom.tld"%>
<t:Test/>

CHAPTER: STRUTS

Why is Struts used?

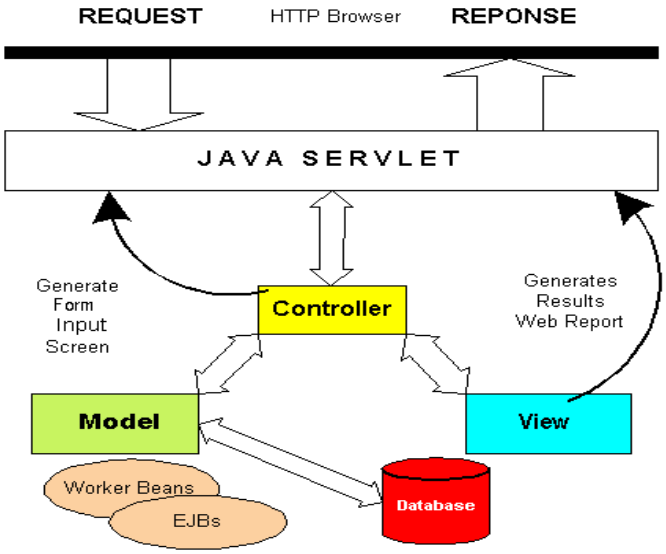
The four most popular and oldest framework which connects to the ground level. There is too much of abstraction for Spring (Popular). It explains how the output and inputs goes through. There is not too much abstraction. It goes through MVC. 40 to 50 % projects are on Struts. Two kinds of struts are 1.x and 2.x. 1.x is more popular and later on Spring became more popular.

What is Struts?

It is a web tier framework. Tier is layer. There are 3-tier. Struts falls on the web tier. It is built on model, view, Controller architecture. It is the communication between the client and the server happens via struts configuration file. It can integrate with other frameworks. Therefore, it is maintainable and extensible.

What is MVC Architecture?

The head of the MVC architecture is the controller. The request goes to the java servlet. In struts we have the default servlet which is the action servlet. Based on the URL it will pass to the controller. Based on the request the call the appropriate the model. The model represents the business logic. Model: Represents data of the application. View: represents the presentation part of application. Controller: control the flow of request and response. All layers word hand in hand. The whole handling of the data, persisting the data, retrieving the data, applying business logic to the data happens in the model. View: it is the representation of the data. View returns to the servlet and gives the response.



What is the Struts Architecture?

It is built on MVC Design Pattern. It uses and extends the Java Servlet API. Some of the key components are Request Handler, Response Handler and Tag Library. Request handler: It is responsible for reading the request and forwarding to the right controller, Response Handler: Taking the response and forwarding to the right view, Tag Library: On the web page to display the data, the inbuilt tag library to communicate from the back end to front end. It separates the business and application data from the presentation layer. The advantages are reusable and expressive. That means you can have same model multiple presentations or same presentations multiple models.

What are the Struts Components?

Controller Components: Action Servlet, Action Class: Struts does not know you are going to call premium. Save action and retrieve action. Request processor: It actually controls the whole flow of request and response.

View Components: JSP, Custom Tags , Java Script

Model Components: Java Classes: All the back end components, the transactional code, which is responsible for the business logic data.

How to configure struts?

- Importing Struts.jar and dependent jars
- Servlet entry in web.xml
- Mention the struts configfiles in the web.xml
- Provide the action mapping in Struts config
- Build the struts configfile
- Create a custom servletif needed

CHAPTER: SPRING

Simple Strut Based Application Code: Web XML

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns="http://java.sun.com/xml/ns/javaee"
	xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
	id="WebApp_ID" version="3.0">
	<display-name>Batch77Struts</display-name>
	<welcome-file-list>
		<welcome-file>home.do</welcome-file>
	</welcome-file-list>

	<servlet>
		<servlet-name>action</servlet-name>
		<servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
		<init-param>
			<param-name>config</param-name>
			<param-value>/WEB-INF/struts-config.xml</param-value>
		</init-param>
	</servlet>

	<servlet-mapping>
		<servlet-name>action</servlet-name>
		<url-pattern>*.do</url-pattern>
	</servlet-mapping>
</web-app>
```

Simple Strut Based Application Code: Struts Configuration

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE struts-config PUBLIC
          "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
          "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">
<struts-config>
	<form-beans>
		<form-bean name="productForm" type="com.struts.form.ProductForm"></form-
bean>
	</form-beans>
	<action-mappings>
```

```
		<action path="/getPrice" name="productForm"
type="com.struts.action.PriceAction">
			<forward name="success" path="/jsp/price.jsp"></forward>
		</action>
		<action path="/home" name="productForm"
type="com.struts.action.HomeAction">
			<forward name="success" path="/jsp/home.jsp"></forward>
		</action>
	</action-mappings>

</struts-config>
```

Simple Strut Based Application Code: Product Form

```
package com.struts.form;

import java.util.List;

import org.apache.struts.action.ActionForm;

public class ProductForm extends ActionForm {
	private String product;
	private Double price;
	private List<String> productList;

	public List<String> getProductList() {
		return productList;
	}

	public void setProductList(List<String> productList) {
		this.productList = productList;
	}

	public String getProduct() {
		return product;
	}

	public void setProduct(String product) {
		this.product = product;
	}

	public Double getPrice() {
		return price;
	}

	public String getPriceStr() {
		return price.toString();
	}
```

```
	public void setPrice(Double price) {
		this.price = price;
	}

}
```

Simple Strut Based Application Code: Price Action

```
package com.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.struts.form.ProductForm;
import com.struts.model.ProductDAO;

public class PriceAction extends Action {
	public ActionForward execute(ActionMapping mapping, ActionForm form,
			HttpServletRequest request, HttpServletResponse response) {
		ProductForm pForm = (ProductForm) form;
		ProductDAO dao = new ProductDAO();
		Double price = dao.getPrice(pForm.getProduct());
		System.out.println("price is : " + price);
		pForm.setPrice(price);
		return mapping.findForward("success");
	}
}
```

Simple Strut Based Application Code: ProductDAO

```
package com.struts.model;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
```

```
import java.util.List;

public class ProductDAO {

	public Double getPrice(String product) {
		Double price = null;
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			PreparedStatement stmt = conn
					.prepareStatement("SELECT price FROM product where
name = ?");
			stmt.setString(1, product);
			ResultSet res = stmt.executeQuery();
			while (res.next()) {
				price = res.getDouble(1);
			}
			res.close();
			stmt.close();
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
		return price;
	}

	public List<String> getProductList() {
		List<String> products = new ArrayList<String>();
		try {
			Class.forName("org.postgresql.Driver");
			Connection conn = DriverManager.getConnection(
					"jdbc:postgresql:hms_77", "postgres", "postgres");
			PreparedStatement stmt = conn
					.prepareStatement("SELECT name FROM product");
			ResultSet res = stmt.executeQuery();
			while (res.next()) {
				products.add(res.getString(1));
			}
			res.close();
			stmt.close();
			conn.close();
		} catch (SQLException e) {
			e.printStackTrace();
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
		return products;
	}
```

```
}
```

Simple Strut Based Application Code: Home JSP

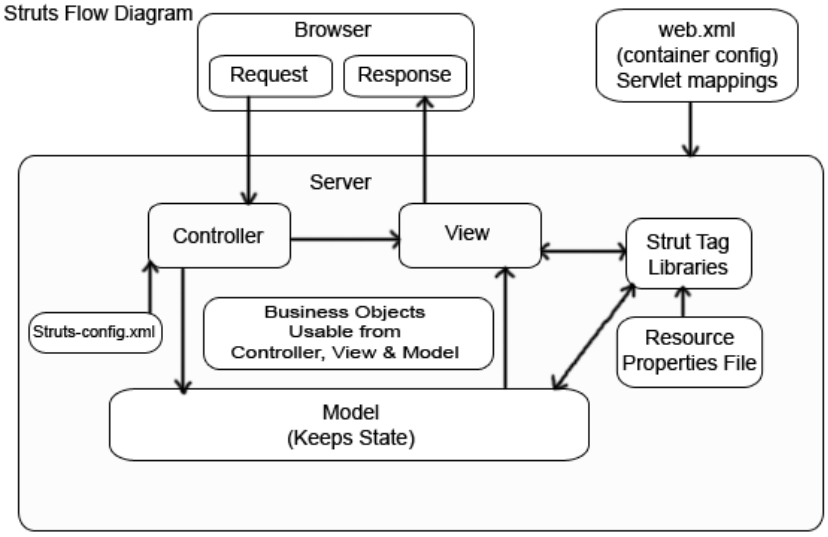
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<html:form action="/getPrice.do" method="get">
		<html:select property="product" name="productForm">
			<html:options property="productList" name="productForm" />
		</html:select>
		<html:submit value="Get Price"></html:submit>
	</html:form>
</body>
</html>
```

Simple Strut Based Application Code: Price JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	The price of the product
```

```
        <bean:write property="product" name="productForm" />
        is :
        <bean:write property="priceStr" name="productForm" />
</body>
</html>
```


What is the Struts Flow Diagram?



At the top is the request which hits the controller through the web.xml. According to the configuration, it hits the controller which hits the action servlet, which involves request processor and the last component is the action. From the action class we will call the model layer. The bean tags were used to write the output to the screen. We can use the resources property files for any messages or resources or some other texts to show it on the screen.

What is your understanding of the Controller?

- org.apache.struts.action. ActionServletis heart of struts framework
- Action Servlet in the web.xml
- Override the action servletif needed
- Provide the url mapping for the servlet
- Mention the required action path in the jsp

Simple Struts Code: Web XML

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns="http://java.sun.com/xml/ns/javaee"
	xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
	id="WebApp_ID" version="3.0">
	<display-name>Batch77Struts</display-name>
	<welcome-file-list>
		<welcome-file>home.do</welcome-file>
	</welcome-file-list>

	<servlet>
		<servlet-name>action</servlet-name>
		<servlet-class>com.struts.servlets.MyCustomActionServlet</servlet-class>
		<init-param>
			<param-name>config</param-name>
			<param-value>/WEB-INF/struts-config.xml</param-value>
		</init-param>
	</servlet>

	<servlet-mapping>
		<servlet-name>action</servlet-name>
		<url-pattern>*.do</url-pattern>
	</servlet-mapping>
</web-app>
```

Simple Struts Code: My Custom Action Servlet

```
package com.struts.servlets;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionServlet;
```

```
public class MyCustomActionServlet extends ActionServlet {
	public void doGet(HttpServletRequest request, HttpServletResponse response)
			throws IOException, ServletException {
		System.out.println("In Do Get");
		super.doGet(request, response);
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
			throws IOException, ServletException {
		System.out.println("In Do Post");
		super.doPost(request, response);
	}
}
```

What are the action class?

It is the key part of the controller. It takes the request from the front end and facilitates the response. It acts as a middle broker or kind of an interface between the view and the model. It can understand what viewer is asking for and convert it to a request which is understood by a model.

Execute method

- Action Mapping: Action servlet will build action mapping. Action Form
- HttpServletRequest and HttpServletResponse
- ActionForward(return type)
- public ActionForwardexecute(ActionMappingmapping, ActionFormform, javax.servlet.http.HttpServletRequestrequest,
- javax.servlet.http.HttpServletResponseresponse) throws java.lang.Exception

Simple Struts Code: My Custom Action Servlet/ Try Catch

```
package com.struts.action;
import java.io.FileWriter;
import java.io.IOException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import com.struts.form.ProductForm;
import com.struts.model.ProductDAO;

public class PriceAction extends Action {
	public ActionForward execute(ActionMapping mapping, ActionForm form,
			HttpServletRequest request, HttpServletResponse response)
			throws IOException {
		ProductForm pForm = (ProductForm) form;
		ProductDAO dao = new ProductDAO();
```

```
			Double price = dao.getPrice(pForm.getProduct());
			System.out.println("price is : " + price);
			pForm.setPrice(price);
			FileWriter f = null;
			try {
				f = new FileWriter("c:/temp/output.txt");
				f.write(price.toString());
			} catch (IOException e) {
				return mapping.findForward("failure");
			} finally {
				f.close();
			}
			return mapping.findForward("success");
		}
}
```

What is action form?

A bean to transfer data from page to Action. You can reset and validate the form and throw error messages. Has on getters & setters for each attribute on the page

Methods: publicvoidreset(ActionMappingmapping,HttpServletRequestrequest)
publicActionErrorsvalidate(ActionMappingmapping,HttpServletRequestrequest)
Struts config declaration: <form-bean name="XForm" type="com.net.XForm"/>

Simple Struts Code: Action form/ Home JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<script type="text/javascript">
	function setSelected(obj) {
		if (obj.value != "") {
			document.getElementById("selected").value = true;
		} else {
			document.getElementById("selected").value = false;
		}
	}
</script>
</head>
<body>
	<html:errors header="errors.header" footer="errors.footer"
		prefix="errors.prefix" suffix="errors.suffix" />
	<html:form action="/getPrice.do" method="get">
		<div>
			<bean:message key="label.product.select" />
			:
```

```
                        <html:select property="product" name="productForm"
                              onchange="setSelected(this)">
                              <html:option value=""></html:option>
                              <html:options property="productList" name="productForm" />
                        </html:select>

                        <html:hidden property="selected" name="productForm"
                              styleId="selected"></html:hidden>
                  </div>
                  <html:reset value="Clear">
                  </html:reset>
                  <html:submit value="Get Price"></html:submit>
            </html:form>
</body>
</html>
```

Simple Struts Code: App Properties

```
label.product.select=select a product

errors.header=<font size="2"><UL>
errors.prefix=<LI><span style="color: red">
errors.suffix=</span></LI>
errors.footer=</UL></font>

error.product.select=The product is required
```

Simple Struts Code: Product Form/ Validate

```
package com.struts.form;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.validator.ValidatorForm;

public class ProductForm extends ValidatorForm {
      private String product;
      private Double price;
      private boolean selected;
      private List<String> productList;

      public List<String> getProductList() {
            return productList;
      }

      public void setProductList(List<String> productList) {
            this.productList = productList;
```

```
	}

	public String getProduct() {
		return product;
	}

	public void setProduct(String product) {
		this.product = product;
	}

	public Double getPrice() {
		return price;
	}

	public boolean isSelected() {
		return selected;
	}

	public void setSelected(boolean selected) {
		this.selected = selected;
	}

	public void setPrice(Double price) {
		this.price = price;
	}

	@Override
	public ActionErrors validate(ActionMapping mapping,
			HttpServletRequest request) {
		ActionErrors errors = new ActionErrors();
		if (!selected) {
			errors.add("InvalidProduct", new ActionMessage(
					"error.product.select"));
		}
		return errors;
	}

	@Override
	public void reset(ActionMapping mapping, HttpServletRequest request) {
		this.product = "";
	}
}
```

Simple Struts Code: Struts-Config

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE struts-config PUBLIC
          "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
          "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">
<struts-config>
	<form-beans>
```

```
            <form-bean name="productForm" type="com.struts.form.ProductForm"></form-
bean>
      </form-beans>
      <action-mappings>
            <action path="/getPrice" name="productForm"
type="com.struts.action.PriceAction"
                  validate="true" input="/jsp/home.jsp">
                  <forward name="success" path="/jsp/price.jsp"></forward>
                  <forward name="failure" path="/jsp/error.jsp"></forward>
            </action>
            <action path="/home" name="productForm"
type="com.struts.action.HomeAction"
                  validate="false" input="/jsp/home.jsp">
                  <forward name="success" path="/jsp/home.jsp"></forward>
            </action>
      </action-mappings>
      <message-resources parameter="app"></message-resources>
</struts-config>
```

What are some of the Struts HTML tags?

TAGLIB Declaration

<%@ tagliburi="/tags/struts-html" prefix="html" %>

TAGS

<html:passwordproperty="prop" size="10"/>
<html:textproperty="text1" size="5"/>
<html:submit>Submit</html:submit>
<html:reset>Reset</html:reset>
<html:errors/>
<html:fileproperty="fileSelectionBox"/>
<html:checkboxproperty="myCheckBox"/>
<html:hiddenproperty="hiddenfield"/>
<html:radiovalue="abc" property="myCheckBox"/>
<html:selectmultiple="true" property="selectBox">
<html:textareaproperty="myTextArea" value="Hello Struts" />
<html:formaction="/Address" method="post">
<html:html>

Simple Struts Code: In Home JSP

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html"%>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<script type="text/javascript">
      function setSelected(obj) {
```

```
			if (obj.value != "") {
				document.getElementById("selected").value = true;
			} else {
				document.getElementById("selected").value = false;
			}
		}
</script>
</head>
<body>
		<html:errors header="errors.header" footer="errors.footer"
			prefix="errors.prefix" suffix="errors.suffix" />
		<html:form action="/getPrice.do" method="get">
			<div>
				<bean:message key="label.product.select" />
				:
				<html:select property="product" name="productForm"
					onchange="setSelected(this)">
					<html:option value=""></html:option>
					<html:options property="productList" name="productForm" />
				</html:select>

				<html:hidden property="selected" name="productForm"
					styleId="selected"></html:hidden>
			</div>
			<html:reset value="Clear">
			</html:reset>
			<html:submit value="Get Price"></html:submit>
		</html:form>
</body>
</html>
```

Simple Struts Code: Product Form

```
package com.struts.form;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.validator.ValidatorForm;

public class ProductForm extends ValidatorForm {
		private String product;
		private Double price;
		private boolean selected;
		private List<String> productList;

		public List<String> getProductList() {
			return productList;
		}

		public void setProductList(List<String> productList) {
			this.productList = productList;
		}
```

```
	public String getProduct() {
		return product;
	}

	public void setProduct(String product) {
		this.product = product;
	}

	public Double getPrice() {
		return price;
	}

	public boolean isSelected() {
		return selected;
	}

	public void setSelected(boolean selected) {
		this.selected = selected;
	}

	public void setPrice(Double price) {
		this.price = price;
	}

	@Override
	public ActionErrors validate(ActionMapping mapping,
			HttpServletRequest request) {
		ActionErrors errors = new ActionErrors();
		if (!selected) {
			errors.add("invalidProduct", new ActionMessage(
					"error.product.select"));
		}
		return errors;
	}

	@Override
	public void reset(ActionMapping mapping, HttpServletRequest request) {
		this.product = "";
	}
}
```


What is Struts validator's framework?

It is used to configure the validation for the forms. The action forms have some field and we can use validator's framework to validate those fields. We don't need to write java code for it. It has some predefined rules or we can make the rules by our self. The struts will validate form beans and will give out any errors if there is any. We need to two XMLs to implement the validation to the form beans. One XML is for rules.XML and the other is actually the real XML. On every JSP they all need to change the body. Most of the time the header and footer remains the same.

What are the struts actions types?
Dispatch Action: To collect related functions in to single action
Lookup Dispatch Action: Similar to Dispatch Action but the methods are mapped using mapClassmethod
Mapping Dispatch Action: Similar to Dispatch action but the methods are managed using multiple action mappings
Others –ForwardAction, IncludeAction, SwitchAction