Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Branch: CSE/IT/MCA Section A/B/C/CCE
Session: 2019-2020
Year: 4th / 3rd Semester: 8th/6th

1st Internship Assignment

Kuldeep Chouhan (Domain: Web Development) Intern **email**: kuldeepchouhan040@gmail.com

Company Name: techbeanssolution pvt. Ltd. **website**: www.techbeanssolution.in
Address: 205, Saraswati Complex A.B. Road Indore
Internal Mentor Name: Amrata Gupta **Date:** 10/02/2020

Question 1- Explain Serialization and Deserialization in java with example ?
Answer- Serialization is a mechanism for converting an object into the bytestream. Deserialization is the reverse process of serialization where the bytestream is used to recreate the actual object in the memory. The bytestream created is platform independent so that it can be deserialized on any platform.
To make the object serializible we have to implement a interface java.io.Serializiable.
Advantages of Serialization

1- To save the state of the object.
2- To transfer an object across a network.

The java.io.Serializable is a marker interface that is it does not contain any method. If a parent class has implemented Serializable interface then child class doesn't need to implement it but vice-versa is not true.

Example (Serilaization)

```
import java.io.*;

class Sample implements Serializable
{
   public int a;
   public String b;

   // Default constructor
   public Sample(int a, String b)
   {
      this.a = a;
      this.b = b;
   }

}

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

```
{
    Sample object = new Sample(1, "Serializationexample");
    String filename = "trial.ser";

    // Serialization
    try
    {
        //Saving of object in a file
        FileOutputStream file = new FileOutputStream(filename);
        ObjectOutputStream out = new ObjectOutputStream(file);

        // Method for serialization of object
        out.writeObject(object);

        out.close();
        file.close();

        System.out.println("Object has been serialized");

    }

    catch(IOException ex)
    {
        System.out.println("IOException");
    }
```

Question 2- Define multithreading in java and why it is used?
Answer-Multithreading in java is a process of executing two or more threads simultaneously to maximum utilization of CPU. Multithreaded applications are where two or more threads run concurrently hence it is also known as Concurrency in Java.

Lifecyle of multithreading

1. New
2. Runnable
3. Running
4. Waiting
5. Dead

New: In this phase, the thread is created using class "Thread class".It remains in this state till the program **starts** the thread. It is also known as born thread.

Runnable: In this page, the instance of the thread is invoked with a start method. The thread control is given to scheduler to finish the execution. It depends on the scheduler, whether to run the thread.

Running: When the thread starts executing, then the state is changed to "running" state. The scheduler selects one thread from the thread pool, and it starts executing in the application.

Waiting: This is the state when a thread has to wait. As there multiple threads are running in the application, there is a need for synchronization between threads. Hence, one thread has to wait, till the other thread gets executed. Therefore, this state is referred as waiting state.

Dead: This is the state when the thread is terminated. The thread is in running state and as soon as it completed processing it is in "dead state".

Some of the commonly used methods for threads are:

start()- This method starts the execution of the thread and JVM calls the run() method on the thread.

Sleep(int milliseconds)- This method makes the thread sleep hence the thread's execution will pause for milliseconds provided and after that, again the thread starts executing. This help in synchronization of the threads.

getName() : This method is used for obtaining a thread's name.

getPriority() :This method is used to obtain a thread's priority.

isAlive() :This method is used to determine whether a thread is still running or not.

run() : This method is the entry point for the thread.

Question 3-Explain in brief about MVC architecture?
Answer-MVC stands for Model, View and Controller. It is a product development architecture. The traditional approach of programming works on Input -> Process -> Output approach while MVC works on Controller → Model -> View. During, traditional approach of programming, the UI coding, business logic and applications data domain was written into a single file which creates lack of maintainability, testability as well as scalability of the application. It helps you create applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

Models- Model objects are the parts of the application that implement the logic for the application's data domain. Often, model objects retrieve and store model state in a database. For example, a Customer object might retrieve information from a database, operate on it, and then write updated information back to a Customer table in a SQL Server database.

Views- Views are the components that display the application's user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Customer table that displays UI Controls based on the current state of a Customer object.

Controllers- Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information the controller handles and responds to user input and interaction.

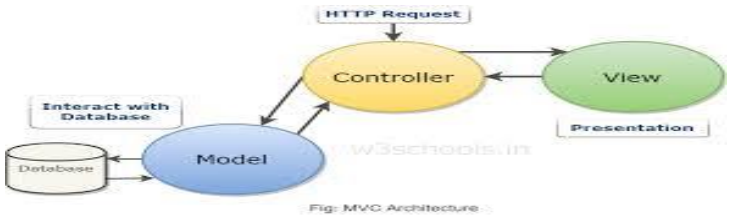

Fig: MVC Architecture

Question4- What is Exception Handling in java? Explain with an example?

Answer- An Exception is an unwanted event that interrupts the normal flow of the program. When an exception occurs program execution gets terminated. There can be several reasons that can cause a program to throw exception. For example: Opening a non-existing file in your program, Network connection problem, bad input data provided by user etc. There are two types of exceptions in java they are checked and unchecked exceptions.

try{}- The try keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally.

catch{}-Java catch block is used to handle the Exception by declaring the type of exception within the parameter.

finally{}- The finally block is used to execute the important code of the program.It is the block which always excutes.

throws-The throws keyword is used to declare exceptions.It is always applied after the methods.

Example

```
import java.io.*;

Class Exceptionhandling
{
  public static void main(String args[])
{
    try{
      int a=50;
      int b=0;
      int c=a/b;
      System.out.println(c);
}
    catch(Exception e)
{
   System.out.println(" Divide by zero");

}

    finally
{
    System.out.println("Exception handling demo");
}

}

}
```

Question5- what is JSON and why it is used?

Answer- JSON stands for Java Script Object Notation. When exchanging data between a browser and a server, the data can only be text. We can also convert any JSON received from the server into JavaScript objects. JSON stores data in the format of key: value pair Since the JSON format is text only, it can easily be sent to and from a server, and used as a data format by any programming language. JSON is an open-standard file format or data interchange format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types.

JSON Syntax Rules

1-Data is in name/value pairs

2-Data is separated by commas

3-Curly braces hold objects

4-Square brackets hold arrays

Example of writing JSON objects

{“firstname”: “Kuldeep” , “lastname”: “chouhan”}

Example of writing JSON array

```
"friends":[
   {"firstName":"Ram", "lastName":"Sharma"},
   {"firstName":"Ravi", "lastName":"Mehta"},

]
```