info@marwadiuniversity.ac.in

Index

Marwadi UNIVERSITY

Name. Shashank Bagda
Subject. OOP
Semester 2

Year. 2021/22
Class. 2TK1
Roll No. 92100133020

No.	Date	Experiment Description / Subject	Page	Marks	Faculty's Signature
1	3/3/22	Worksheet - 1			
2	17/3	Worksheet - 3			
3	30/3	Worksheet - 4			
4	7/4	Worksheet - 6			
5	14/4	Worksheet - 5			
6	31/5	Worksheet - 2			
7	31/5	Worksheet - 7 to 10			

Subject Code: 01CT0105

Subject Name: Object Oriented Programming

B. Tech. Year – I (Semester II)

UNIT – 2

Worksheet – 2

Enrollment No: 92100133020

Name: Shashank Bagda

Subject Faculty:

Prof. Kapil Shukla

Information and Communication Technology Department,

Marwadi University

Q-1: Explain Brief History of JAVA.

Answer: Java is general purpose, object oriented programming language developed by Sun Microsystems of USA in 1991. Originally called by James Gosling, one of the inventors of the language. Java was designed for the development of software for consumer like electronic devices like TVs, VCRs, and such other machines. The goal had a strong impact on the development team to make the language simple, portable & highly reliable.

Q-2: List and explain features of Java.

Answer: The inventors of Java wanted to design a language which could offer solution to some of the problems encounted in modern programming. They wanted the language to be not only reliable, portable and distributed but also simple, compact and interactive. Jave describes as...

Compiled & Interpreted:

Usually a computer language is either compiled or interpreted. Java combines both these approaches thus making Java a two-stage system. First, Java compiler translated source code into what is known as bytecode instruction. We can say that Jave is both a complied & an interpreted language.

Platform - Independent & Portable:

The most significant contribution of Java over

Other language is its portability. Java programs can be easily moved from one computer system to another, anywhere & anytime. Changes and upgrades in operating system, processors & system resources will not force any changes in Java programs. This is the reason why Java has become a popular language for programming on Internet which interconnects different kinds of systems worldwide. We can download a Java applet from a remote computer onto our local system via internet and execute it.

Object Oriented :

Java is a true object-oriented language. Almost everything in Java is an object. All program code and data reside within objects and classes. Java comes with an extensive set of classes, arranged in packages, that we can use in our program by inheritance. The object model in Java is simple & easy to extend.

Distributed :

Java is designed as a distributed language for creating applications on networks. It has the ability to share both data and programs. Java applications can open and access remote objects on Internet as easily as they can do in a local system. This enables multiple programmers at multiple remote locations to collaborate and work together on a single project.

Q 3: How Java is different than C and C++? List it.

Answer: Java & C

Java is an object oriented language and has mechanism to define classes & objects. In an efforts to build a simple and safe language, The Java team did not include some of the C feature in Java.

1) Java does not include the C unique statement keywords size of and typedef.
2) Java does not contain the data types struct & union.
3) Java does not define the type modifiers keywords auto, extern, register, signed & unsigned.
4) Java does not support an explicit pointer type.
5) Java does not have a preprocessor & therefore we cannot use # define, # include & # ifdef.

Java & C++

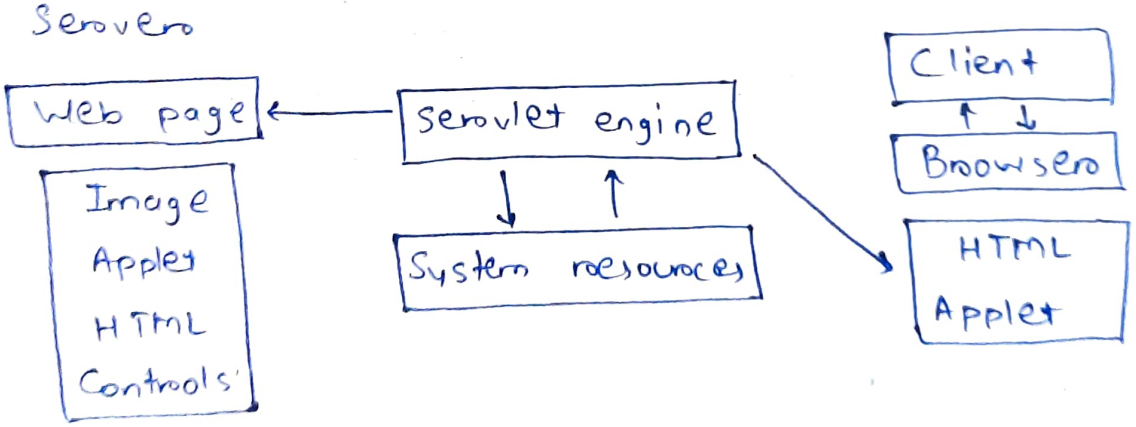
Java is a True object - oriented language while C++ is basically C with object - oriented language extension.

1) Java does not support operator overloading
2) Java does not have templat classes as in C++
3) Java does not support multiple inheritance of classes. This is accomplished using a "interface"
4) Java does not use pointer.
5) Java has replaced the destructor function with a finalize function.

Q-6: Explain Applet and Servlet. [With Diagram]

Answer:

- Applet is a small applications which is written in Java and delivered to users in form of byte code.
- Applets are executed on client side applets are used to provide interactive features to web applications that can't be provided by HTML like mouse input, etc.
- Package available in Applets are : java.applet.*
- Servlet in a Java class used to extend capabilities of a servers.
- Servlets are executed on servers side servlets are the Java Front counterpart to other dynamic web content technology such as PHP & Asp.NET
- Life cycle of servers are servlet() & destroy();



Answer:

1) Eclipse : Supports Java, C, C++, Perl, Python, etc.
2) Netbeans : It is a framework that is used for the development of Java Swing desktop applications
3) JUnit : It is an open source Testing framework which has linked JAR at compile time. It is used to create Test cases.
4) Apache Spark : It is use to speed up processing jobs in Hadoop Systems.
5) Jenkins : It is used for testing & is an open source framework which is written in Java programming language.
6) Android Studio : It is used for developing apps on every type of Android device.

Q-6: List commonly used packages from Java Standard Library.

Answer:

1. Java Lang package
2. Java Util package
3. Java Time package
4. Java IO package
5. Java net package
6. Java NIO package
7. Java SQL Package
8. Java XML* package
9. Java FX* package
10. Java Swing Package

Q-7: Explain following Java Code in your own words.

```
class Test
{
    public static void main(String str[])
    {
        System.out.println("ICT Department");
    }
}
```

Answer:

⇒ class Test

Here 'class' is the keyword to declare class. & 'Test' is the name of the declared class.

⇒ public static void main (string str[])

Here 'public' is access modifiers i.e it is visible to everyone. Static keyword is used to declare any method as static. Void is the return type of the method. String str[] is used for command line argument.

⇒ System.out.println ("ICT Department");

Here system.out.println is used to print statement & it will return "ICT Department".

Q 8: Explain Java Program Structure with figure.

Answer:

A java program may contain one or more section as shown below

Documentation Section ⟶ Suggested

Package Statement ⟶ Optional

Import statements ⟶ Optional

Interface Statement ⟶ Optional

Class Definitions ⟶ Optional

Main method class
{
  Main method Definition
}
⟶ Essential

Documentation section:

The documentation section comprises a set of comment lines giving the name of the prog. the author & others details. comments must explain why and what of classes and how of algorithms. /*....*/ known as documentation comment. This forom of comment is used foro generating documentation automatically.

Package Statement:

The first statement allowed in a java file is a package statement. This statement declares a package name & inform the compiler that the classes defined here belong to this package

Import statement:

The next thing after package and before class definition should be a number of import statements.

Interface Statement :

An interface is like a class but include a group of method declarations. This is also an optional section and is used only when we wish to implement the multiple interface feature in the program.

Class Definition :

A java program may contain multiple class definations. Classes are the primary & essential elements of a java program. These classes are used to map the objects of real-world problems. The number of classes used depends on the complexity of the problem.

Main method Class :

A simple java program may contain only this part. The main method creates objects of various classes & establishes communication betn them.

Q-9: Explain Java Program Compilation and execution Process with command.

Answer:

```
public class Test
{
  public static void main (String args[])
  {
    System.out.println("Hello World");
  }
}
```

Save the above code with file name as: Test.java. Open cmd window and go to the directory where you saved the file. Type javac Test.java. It will create a class file. Now write: java Test in cmd, so it will print Hello World on cmd prompt.

Interface Statement ?

An interface is like a class but include a group of method declarations. This is also an optional section and is used only when we wish to implement the multiple interface feature in the program.

Class Definition :

A java program may contain multiple class definations. Classes are the primary & essential elements of a java program. These classes are used to map the objects of real-world problems. The number of classes used depends on the complexity of the problem.

Main method Class :

A simple java program may contain only this part. The main method creates objects of various classes & establishes communication betn them.

Q-9: Explain Java Program Compilation and execution Process with command.

Answer:

```
public class Test
{
    public static void main (String args[])
    {
        System.out.println ("Hello World");
    }
}
```

Save the above code with file name as : Test.java. Open cmd window and go to the directory where you saved the file. Type javac Test.java . It will create a class file. Now write : java Test in cmd, so it will print Hello World on cmd prompt.

Eg:

```
D:\> javac Test.java
D:\> java Test
Hello World.
```

Q-10: Write a Short note on JVM.

Answer: JVM is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed. JVMs are available for many hardware and software platforms. JVM performs following operations:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM provides definitions for the

- memory area
- class file format
- register set
- garbage - collected heap
- Fatal error reporting etc.

References

Q No	Book Name	Page No
1	Proogroamming with Java	11
2	Proogroammming with Java	12 - 14
3	Proogroammming with Java	15, 16
4	Java Complete	225, 617
5	Othere OOP Reference	
6	Proogroammming with Java	384
7	By Own	
8	Proogroammming with Java	29, 30
9	Proogroammming with Java	37, 38
10	Proogroammming with Java	38