



Victoria University of Bangladesh

58/11/A, Panthapath, Dhaka, Bangladesh

Object Oriented Programming

CSI-211

Final Assessment

Submitted By:

ARS Nuray Alam Parash

ID # 2120180041

18th Batch, BSc in CSE

E-mail: chatokparash@gmail.com

Mobile: 01922339393

Submission Date: 5 February 2023

Submitted To:

Umme Khadiza Tithi

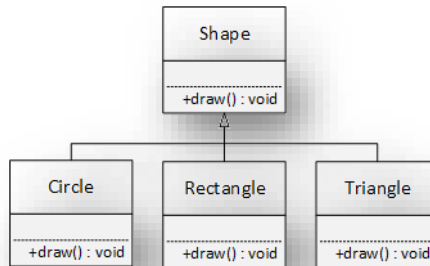
Lecturer

Department of Computer Science &
Engineering

Victoria University of Bangladesh (VUB)

Answer to the Question No- 1 (a)

Polymorphism: Refers to the ability of a variable, object, or function to take on multiple forms. For example, the verb run has a different meaning in English if you use it with a laptop, a foot race, and business. Here, we understand the importance of running based on the other words used along with it. The same also applies to Polymorphism.



Answer to the Question No- 1 (b)

Class in JAVA: A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

- **Modifiers:** A class can be public or has default access (Refer this for details).
- **Class name:** The name should begin with a initial letter (capitalized by convention).
- **Superclass (if any):** The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
- **Interfaces (if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.

Objects in JAVA: It is a basic unit of Object-Oriented Programming and represents real-life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of:

- **State:** It is represented by attributes of an object. It also reflects the properties of an object.

- **Behavior:** It is represented by the methods of an object. It also reflects the response of an object to other objects.
- **Identity:** It gives a unique name to an object and enables one object to interact with other objects.

Answer to the Question No- 1 (c)

Key benefits of Object-Oriented Programming:

- Through inheritance, we can eliminate redundant code and extend the use of existing classes.
- We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch. This leads to saving development time and higher productivity.
- The principle of data binding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
- It is possible to map objects in the problem domain to those objects in the program.
- It is possible to have multiple objects coexist without any interference.
- It is easy to partition the work in a project based on objects.
- The data-centered design approach enables us to capture more details of a model in an implementable form.
- Object-Oriented systems can be easily upgraded from small to large systems.
- Message-passing techniques for communication between objects make the interface descriptions with external systems much simpler.
- Software complicity can be easily managed.

Answer to the Question No- 1 (d)

Print Image:



IDE Image:

```
love_java.java x Mypanel.java
1
2 import java.awt.Color;
3 import javax.swing.JFrame;
4
5 public class love_java extends JFrame {
6
7     love_java() {
8         setSize(800,600);
9         setDefaultCloseOperation(EXIT_ON_CLOSE);
10        setLocationRelativeTo(null);
11
12        Mypanel panel=new Mypanel();
13        panel.setBackground(Color.black);
14        add(panel);
15
16    }
17
18    public static void main(String[] args) {
19        // TODO Auto-generated method stub
20        new love_java().setVisible(true);
21    }
22 }
23
24
25
```

```
love_java.java x *Mypanel.java x
1
2 import java.awt.Color;
3 import java.awt.Font;
4 import java.awt.Graphics;
5 import java.awt.Image;
6 import javax.swing.ImageIcon;
7 import javax.swing.JPanel;
8
9 public class Mypanel extends JPanel {
10
11     @Override
12     public void paint(Graphics g) {
13         super.paint(g);
14
15         g.setColor(Color.white);
16         g.setFont(new Font(Font.SANS_SERIF, Font.BOLD,50));
17         g.drawString("print java program", 50, 50);
18
19         String url="D:\\VUB\\9th Semester\\CSI-211_Object Oriented Programming\\VUB.jpg";
20         Image image=new ImageIcon(url).getImage();
21         g.drawImage (image, 50, 100, 640, 480, null);
22     }
23 }
24
25
```

Answer to the Question No- 2 (a)

JAVA Applets: JAVA Applets are another type of Java program that are implemented on Internet browsers and are always run as part of a web document.

It is a special type of program that is embedded in the webpage to generate dynamic content. It runs inside the browser and works on the client side.

Answer to the Question No- 2 (b)

Technology is constantly going through an evolution and so are the languages that are used to develop them. Java is one of the popular programming languages having a number of applications.

- **Mobile Applications:** Java is considered the official programming language for mobile app development. It is compatible with software such as Android Studio and Kotlin. Now you must be wondering why only Java? The reason is that it can run on Java Virtual Machine (JVM), whereas Android uses DVM (Dalvik Virtual Machine) to execute class files. These files are further bundled as an Android application package (APK). With Java and its OOPs principles, it provides better security and ease of simplicity with Android.

- **Desktop GUI Applications:** All desktop applications can easily be developed in Java. Java also provides GUI development capability through various means mainly Abstract Windowing Toolkit (AWT), Swing, and JavaFX. While AWT holds a number of pre-assembled components like a menu, list, and button. Swing is a GUI widget toolkit, it provides certain advanced elements like trees, scroll panes, tables, tabbed panels, and lists.

- **Web-based Applications:** Java is also used to develop web applications. It provides vast support for web applications through Servlets, Struts, or JSPs. With the help of these technologies, you can develop any kind of web application that you require. The easy coding and high security offered by this programming language allow the development of a large number of applications for health, social security, education, and insurance.

- **Enterprise Applications:** Java is the first choice of many software developers for writing applications and Java Enterprise Edition (Java EE) is a very popular platform that provides API and runtime environment for scripting. It also includes network applications and web services. JavaEE is also considered the backbone for a variety of banking applications that have Java running on the UI to the back-server end.

- **Scientific Applications:** Software developers see Java as the weapon of choice when it comes to coding scientific calculations and mathematical operations. These programs are designed

to be highly secure and lightning fast. they support a higher degree of portability and offer low maintenance. Some of the most powerful applications like MATLAB use Java for interacting with the user interface as well as part of the core system.

- **Gaming Applications:** Java has the support of the open-source most powerful 3D-Engine, the jMonkeyEngine which has unparalleled capability when it comes to the designing of 3D games. However, it does cause an occasional latency issue for games as garbage collection cycles can cause noticeable pauses. This issue will be solved in the newer versions of JVMs.

- **Big Data technologies:** Java is the reason why the leading Big Data technologies like Hadoop have become a reality and also the most powerful programming languages like Scala are existing. It is crystal clear that Java is the backbone when it comes to developing Big Data using Java.

- **Business Applications:** Java EE platform is designed to help developers create large-scale, multi-tiered, scalable, reliable, and secure network applications. These applications are designed to solve the problems encountered by large enterprises. The features that make enterprise applications powerful, like security and reliability, often make these applications complex. The Java EE platform reduces the complexity of enterprise application development by providing a development model, API, and runtime environment that allow developers to concentrate on functionality.

- **Distributed Applications:** Distributed applications have several common requirements that arise specifically because of their distributed nature and of the dynamic nature of the system and platforms they operate on. Java offers options to realize these applications. The Jini (Java Intelligent Networking Infrastructure) represents an infrastructure to provide, register, and find distributed services based on its specification. One integral part of Jini is JavaSpaces, a mechanism that supports the distribution, persistence, and migration of objects in a network.

- **Cloud-Based Applications:** Cloud computing means the on-demand delivery of IT resources via the internet with pay-as-you-go pricing. It provides a solution for IT infrastructure at a low cost. Java provides you with features that can help you build applications meaning that it can be used in SaaS, IaaS, and PaaS development. It can serve the companies to build their applications remotely or help companies share data with others, whatever the requirement.

Answer to the Question No- 2 (c)

Multithreading: It is a process of executing multiple threads simultaneously.

A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

However, we use multithreading than multiprocessing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

Java Multithreading is mostly used in games, animation, etc.

Common Advantages of Multithreading:

- Enhanced performance by decreased development time
- Simplified and streamlined program coding
- Improved GUI responsiveness
- Simultaneous and parallelized occurrence of tasks
- Better use of cache storage by utilization of resources
- Decreased cost of maintenance
- Better use of CPU resource

Common Disadvantages Multithreading:

- Complex debugging and testing processes
- Overhead switching of context
- Increased potential for deadlock occurrence
- Increased difficulty level in writing a program
- Unpredictable results

Answer to the Question No- 2 (d)

RMI stands for Remote Method Invocation. It is a mechanism that allows an object residing in one system (JVM) to access/invoke an object running on another JVM.

RMI is used to build distributed applications; it provides remote communication between Java programs. It is provided in the package java.rmi.

Difference between stub and skeleton in RMI-

stub

The stub is an object, that acts as a gateway for the client side. All the outgoing requests are routed through it. It resides on the client side and represents the remote object. When the caller invokes the method on the stub object, it does the following tasks:

- It initiates a connection with a remote Virtual Machine (JVM),
- It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM),
- It waits for the result
- It reads (unmarshals) the return value or exception, and
- It finally, returns the value to the caller.

skeleton

The skeleton is an object, that acts as a gateway for the server-side object. All incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:

- It reads the parameter for the remote method
- It invokes the method on the actual remote object, and
- It writes and transmits (marshals) the result to the caller.

In the Java 2 SDK, a stub protocol was introduced that eliminates the need for skeletons.

Answer to the Question No- 3 (a)

Servlet technology is used to create a web application (resides on the server side and generates a dynamic web page).

Servlet technology is robust and scalable because of the java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

Servlet Advantage

- Servlets provide a way to generate dynamic documents that is both easier to write and faster to run.
- provide all the powerful features of JAVA, such as Exception handling and garbage collection.
- Servlet enables easy portability across Web Servers.
- Servlet can communicate with different servlets and servers.
- Since all web applications are stateless protocols, servlet uses its own API to maintain session

Servlet Disadvantage

- Designing in servlet is difficult and slows down the application.
- Writing complex business logic makes the application difficult to understand.
- You need a Java Runtime Environment on the server to run servlets. CGI is a completely language-independent protocol, so you can write CGIs in whatever languages you have available (including Java if you want to).

[Answer to the Question No- 3 \(b\)](#)

Check Whether a Number is Even or Odd in Java

Given an integer input the objective is to write a Java code to Check Whether a Number is Even or Odd. To do so the main idea is to divide the number by 2 and check if it's divisible or not. It's an Even number if it's perfectly divisible by 2 or an Odd number otherwise.

Here are the Methods to solve the above-mentioned problem,

Method 1: Using Brute Force

Method 2: Using Ternary Operator

Method 3: Using Bitwise Operators

Method 1: Using Brute Force

This method simply checks if the given input integer is divisible by 2 or not. If it's divisible then print Even or Odd otherwise.

Algorithm-

The working of the above code is mentioned below

1. Input an integer input "**number**"
2. Check whether the number is divisible by 2
3. This means using modulo/remainder operator leaves 0 as a remainder
4. Do: if (number % 2 == 0)
 1. if yes, print "Even number"
 2. if not, print "Odd number"

Java Code-

```
Code Java (OpenJDK 13.0.1) Output Panel Run Code
```

```
1 public class Main
2 {
3     public static void main(String[] args) {
4         int number = 29;
5
6
7         //checking whether the number is even or odd
8         if (number % 2 == 0)
9             System.out.println(number + " is Even");
10        else
11            System.out.println(number + " is odd");
12    }
13 }
```

29 is odd

Input Panel

Method 2: Using Ternary Operator

This Method uses the ternary operator to check if the integer input is divisible by 2, If true print Even or Odd otherwise.

Algorithm-

The working of the above code is as follows,

1. Input an integer input “**number**”
2. Check whether the number is divisible by 2 using the ternary operator
3. $(\text{number} \% 2) ? (\text{cout} \ll \text{“Even”}) : (\text{cout} \ll \text{“Odd”})$

Java Code-

```
Code Java (OpenJDK 13.0.1) Output Panel Run Code
```

```
1 public class Main
2 {
3     public static void main (String[]args)
4     {
5         int number = 29;
6
7         //Checking if the number is divisible by 2
8         String status = number % 2 == 0 ? " is Even" : " is Odd";
9         System.out.println (number + status);
10    }
11 }
```

29 is Odd

Input Panel

>> END <<