UNIT 1 -

Q. 1 Explain various datatype supported in JAVA

Q. 2 Write for, do while, while loop to compute the following sum

4+ 8 + 12 + 16 + ----------+ 80

Q. 3 Explain type conversion and casting with example.

Q. 4 WAP to print sum of square of first 20 even numbers.

Q. 5 Explain basic features of JAVA

Q. 6 What is operator? Explain operator precedence & Associativity Rule

Q. 7 Explain different types of operator

Q. 8 WAP to find largest among 3 numbers using if statement & ternary operator

Q. 9 Explain steps for executing java program. What are source and byte code in java?

Q. 10 Difference between OOP vs POP

Q. 11 WAP to compute result of 1*3*5* ---------- * 25

UNIT 2 -

Q. 1 Write a program to find the largest among three integer numbers by using command line arguments.

Q. 2 Write a program to calculate area and perimeter of rectangle using constructor.

Q. 3 Explain difference between local variable, class variable, instance variable

Q. 4 Explain the use of this keyword with suitable example.

Q. 5 What is constructor? Explain different types of constructor with examples.

Q. 6. What is constructor? Explain constructor overloading in java with example.

Q. 7 What is array? How to pass one dimensional array to method with example.

Q. 8 Write a program to out sum and average of any five numbers using command line argument.

Q. 9. Explain garbage collection in java

Q. 10. Explain static variable (class variable) & static method (class method) with examples.

Q. 11 Write a program to print sum of all elements in array of 10 integer numbers.

UNIT 3 -

1. **Explain package in java. Explain steps for creating user defined package in java**
2. **Explain about enum type in java with example**
3. **What method overloading in java with example**
4. **What is method overriding in java with example**
5. Difference between method overloading and method overriding OR compile time vs run time polymorphism OR early binding vs late binding.
6. **Explain Abstract class & abstract method in java**
7. **Explain final variable, final class, final method in java with example**
8. Explain super keyword with example
9. Explain interface with example. How it is differed from abstract classes.
10. Explain Inheritance with different types with examples.

UNIT 4 -

1. **Explain about BufferedInputStream and BufferedOutputStream with example**
2. Difference between checked vs unchecked exception
3. Difference between throw and throws keyword
4. **Write a java program to write data into FileOutputStream**
5. **What is Exception handling mechanism? Write an application which handles Arithmetic and ArrayIndexOutOfBoundsException**
6. **List out various methods associated with file objects and explain with examples.**
7. Write a java program to write data into file using FileWriter class
8. **Write a java program to handle multiple exceptions**
9. Write a java program to handle NullPointerException in java
10. **Write a program to implement FileInputStream and FileReader class.**
11. **Explain try, catch, throw, finally, throws keyword.**
12. **What is user defined exception? Write a program to handle user defined exception.**
13. **Write a program to implement PrintWriter class**

UNIT 5 -

Q. 1 Write an applet which displays the message that illustrates order in which init(), start(), stop(), and paint() methods are called when an applet starts up.

Q. 2 Explain lifecycle of applet

Q. 3 Write a program that shows screen shot with applet running inside the applet viewer. The applet should display your name.

Q. 4 Write notes on i) getDocumentBase() ii) getCodeBase()

Q. 5 What is applet? Difference between applet and application

Q. 6 Write an applet to draw big oval and two small oval inside it. Set different color to each component and set background color as black

Q. 7 Write an applet to draw rectangle, rounded rectangle and fill rectangle, oval and fill oval in the applet.

Q. 8 Write a applet to draw rectangle using drawLine(). Add “Welcome” string inside rectangle.

Q. 9. Explain Graphics class & Color class with example in JAVA.

Q. 10 Write a simple applet to draw filled circle within outlines rounded rectangle.

Q. 11. Explain font class in applet

Q. 12. Explain font metric in applet

Q. 13 Explain applet html tag in JAVA

UNIT 6 -

1. **Explain procedure for creating Menu, Menubar, MenuItem.**
2. **What do you mean by event delegation model in java**
3. Compare various layout managers in brief.
4. **What do you mean by adapter class? Explain with examples**
5. How do the event objects register the event listener in java.
6. What are different types of AWT components? How are these AWT Components added to containers.
7. **Write a program to implement all the methods of mouse listener and mouse motion listener**.
8. **Write a simple program that accepts two integer numbers via text field object, after pressing (add) button, the sum numbers should displayed in text area object.**
9. Write a program to create button “ok”, when user press button, text should change to “its ok”.
10. Write a simple program that accepts radius via text field object, after pressing (calculate) button, area of circle should display.

/*Q. 2 Write for, do while, while loop to compute the following sum

4+ 8 + 12 + 16 + ----------+ 80*/

```
class Example
{
public static void main(String args[])
{
int s = 0;
for(int i=4;i<=80;i=i+4)
{
s = s + i;
}
System.out.println("Sum = " +s);
}
}
```

/*Q. 4 WAP to print sum of square of first 20 even numbers.*/

```
class Example
{
public static void main(String args[])
{
int s = 0;
for(int i=2;i<=40;i=i+2)
{
s = s + i;
}
System.out.println("Sum = " +s);
}
}
```

//WAP to find largest among 3 numbers using if statement & ternary operator.

// Using if statement –

```
import java.util.Scanner;
class Example
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter first number - ");
		int x = sc.nextInt();
		System.out.println("Enter second number - ");
		int y = sc.nextInt();
		System.out.println("Enter third number - ");
		int z = sc.nextInt();

		if(x>y && x>z)
		{
			System.out.println(x + " is largest number”);
		}
		else if(y>z)
		{
			System.out.println(y + “is largest number”);

		}
		else {
			System.out.println(z + " is largest number”);
		}
	}
}
```

Output -

Enter first number - 15

Enter second number - 10

Enter third number - 8

15 is greater than 10 and 8

```
//Using ternary operator
class Example
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter first number - ");
		int x = sc.nextInt();
		System.out.println("Enter second number - ");
		int y = sc.nextInt();
		System.out.println("Enter third number - ");
		int z = sc.nextInt();

		int big = (x>y&&x>z) ? x : (y>z) ? y : z;

		System.out.println("Largest number is " +big);
	}
}
```

Output -

Enter first number - 15

Enter second number - 10

Enter third number - 8

15 is largest number

//WAP to compute result of 1*3*5* ---------- * 25 –

```
class Example
{
	int product = 1;
	public static void main(String args[])
	{
		for(int i=1;i<=25;i=i+2)
		{
			product = product * i;
		}
		System.out.println("Result = " +product);
	}
}
```

// Q. 1 Write a program to find the largest among three integer numbers by using command line arguments.

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

		int x = Integer.parseInt(args[0]);
		int y = Integer.parseInt(args[1]);
		int z = Integer.parseInt(args[2]);

		if(x>y && x>z)
		{
			System.out.println(x + " is largest number");
		}
		else if(y>z)
		{
			System.out.println(y + " is largest number");
		}
		else
```

```
            {
                System.out.println(z + " is largest number");
            }
        }
}
```

Output -

Compile - javac -d . Example.java

Run - java Example 15 10 8

15 is largest number

// Q. 2 Write a program to calculate area and perimeter of rectangle using constructor.

```
import java.util.Scanner;
class Rectangle
{

    Rectangle()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Length -");
        int l = sc.nextInt();
        System.out.println("Enter Width -");
        int w = sc.nextInt();
    }
    void area()
    {
        System.out.println("Area of Rectangle = "+ (l*w));
    }
    void perimeter()
    {
        System.out.println("Perimter of Rectangle = "+ (2*(l+w)));
    }
}
```

```
class Example
{
	public static void main(String[] args)
	{
		Rectangle r = new Rectangle();
		r.area();
		r.perimeter();
	}
}
```

Output -

```
Enter length - 10
Enter width - 5
Area of Rectangle = 50
Perimeter of Rectangle = 30
```

Unit 4 –

1.

Below is a simple Java program that uses `BufferedInputStream` and `BufferedOutputStream` to copy the contents of one file to another. These classes are used to improve the performance of reading and writing data by using buffered I/O operations.

```java
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileCopyWithBufferedStreams {
    public static void main(String[] args) {
        String sourceFilePath = "path/to/source/file.txt";
        String destinationFilePath = "path/to/destination/file.txt";

        try (
```

```
        BufferedInputStream bufferedInputStream = new BufferedInputStream(new
FileInputStream(sourceFilePath));
        BufferedOutputStream bufferedOutputStream = new BufferedOutputStream(new
FileOutputStream(destinationFilePath))
    ) {
        // Create a buffer to read and write data efficiently
        byte[] buffer = new byte[1024];
        int bytesRead;

        // Read from the source file and write to the destination file
        while ((bytesRead = bufferedInputStream.read(buffer)) != -1) {
            bufferedOutputStream.write(buffer, 0, bytesRead);
        }

        System.out.println("File copied successfully.");

    } catch (IOException e) {
        e.printStackTrace();
    }
  }
}
```

In this program:

- `BufferedInputStream` is used to read data efficiently from a file.

- `BufferedOutputStream` is used to write data efficiently to another file.

- A buffer of size 1024 bytes is used to read and write data in chunks, which can improve the overall performance.

- The program uses a try-with-resources statement to automatically close the streams when they are no longer needed.

Make sure to replace the placeholder paths (`"path/to/source/file.txt"` and `"path/to/destination/file.txt"`) with the actual paths of your source and destination files.

Q. 13

```
import java.io.File;
import java.io.PrintWriter;
public class PrintWriterExample {
```

```
    public static void main(String[] args) throws Exception {
//Data to write in File using PrintWriter
      PrintWriter writer = new PrintWriter(new File("D:\\testout.txt"));
        writer.write("Hello world");
        writer.flush();
        writer.close();
    }
}
```

Unit 5 -

Q. 1 Write an applet which displays the message that illustrates order in which init(), start(), stop(), and paint() methods are called when an applet starts up.

```
// <applet code="AppletLifecycleDemo.class" width="300" height="100"></applet>

import java.applet.Applet;
import java.awt.Graphics;

public class AppletLifecycleDemo extends Applet {
    public void init() {
        System.out.println("init() method called");
    }

    public void start() {
        System.out.println("start() method called");
    }
    public void stop() {
        System.out.println("stop() method called");
    }


    public void paint(Graphics g) {
        System.out.println("paint() method called");
        g.drawString("Applet Lifecycle Demo", 20, 20);
```

```
    }
}
```

Console Output -

```
init() method called
start() method called
stop() method called
start() method called
paint() method called
```

Q. 3 Write a program that shows screen shot with applet running inside the applet viewer. The applet should display your name.

```
// <applet code="Example.class" width="300" height="100"></applet>

import java.applet.Applet;
import java.awt.Graphics;

public class Example extends Applet {

    public void paint(Graphics g) {
        g.drawString("Your name");
    }
}
```

Q. 6 Write an applet to draw big oval and two small oval inside it. Set different color to each component and set background color as black

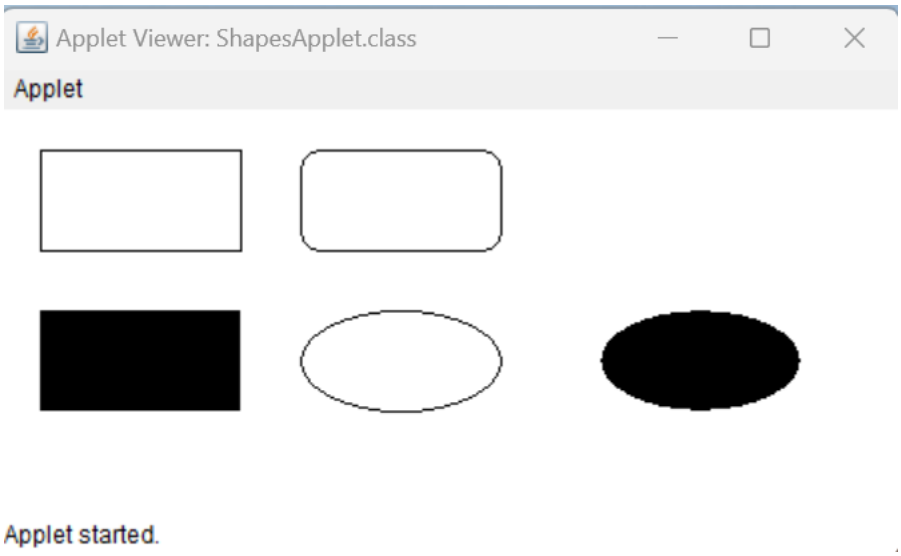
```
// <applet code="Example.class" width="300" height="100"></applet>

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
```

```
public class Example extends Applet {
    public void init() {
        setBackground(Color.black);
    }
    public void paint(Graphics g) {
        // Set colors for the ovals
        g.setColor(Color.red);
        g.fillOval(50, 50, 200, 150); // Big oval

        g.setColor(Color.green);
        g.fillOval(100, 80, 40, 30); // Small oval 1

        g.setColor(Color.blue);
        g.fillOval(160, 80, 40, 30); // Small oval 2
    }
}
```

Q. 7 Write an applet to draw rectangle, rounded rectangle and fill rectangle, oval and fill oval in the applet.

```
// <applet code="ShapesApplet.class" width="450" height="200"></applet>

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class ShapesApplet extends Applet {

    public void paint(Graphics g) {
        // Draw a rectangle
        g.drawRect(20, 20, 100, 50);

        // Draw a rounded rectangle
        g.drawRoundRect(150, 20, 100, 50, 20, 20);

        // Draw a filled rectangle
```

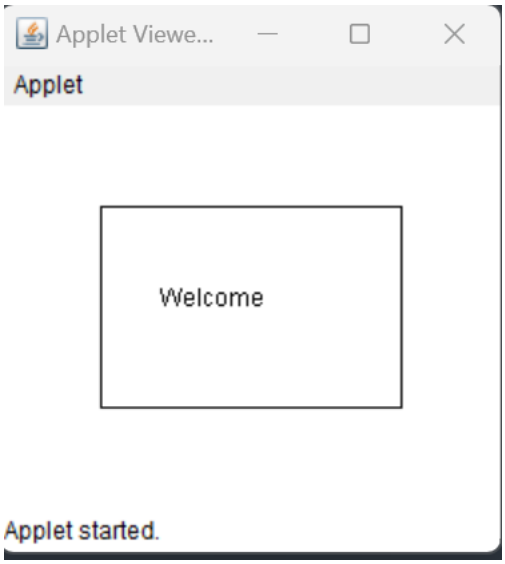
```
        g.fillRect(20, 100, 100, 50);

        // Draw an oval
        g.drawOval(150, 100, 100, 50);

        // Draw a filled oval
        g.fillOval(300, 100, 100, 50);
    }
}
```



Q. 8 Write a applet to draw rectangle using drawLine(). Add “Welcome” string inside rectangle.

```
// <applet code="RectangleWithTextApplet.class" width="250" height="200"></applet>

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class RectangleWithTextApplet extends Applet {

    public void paint(Graphics g) {
```

```
        // Draw the rectangle using drawLine()
        g.drawLine(50, 50, 200, 50);
        g.drawLine(200, 50, 200, 150);
        g.drawLine(200, 150, 50, 150);
        g.drawLine(50, 150, 50, 50);

        // Add the "Welcome" string inside the rectangle
        g.drawString("Welcome", 80, 100);
    }
}
```



Q. 10 Write a simple applet to draw filled circle within outlines rounded rectangle.

```
// <applet code="ShapesApplet.class" width="250" height="200"></applet>

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class ShapesApplet extends Applet {

    public void paint(Graphics g) {
```

```
        // Draw the rounded rectangle outline
        g.drawRoundRect(50, 50, 200, 150, 30, 30);

        // Draw the filled circle inside the rounded rectangle
        g.fillOval(50, 50, 200, 150);
    }
}
```