Name : Jain Apurva Sanjay **Department : M .C. A (Semester - 1)**
Roll No : 12 **Division : A**
Subject : OOPJ **Subject Code : (MTCA13104**

SARVAJANIK UNIVERSITY Sarvajanik College of engineering and technology Masters Of Computer Applications, SURAT

Subject Name:OOPJ(MTCA13104) **Academic Year: 2023-2024**

INDEX

Sr. No	Definition	Page No.	Date	Signature
1.	Install JDK and Write a Program to print Hello World on the console			
2.	Write an application which takes input from command line and convert into integer using parseInt()			
3.	Write a Java program to check whether a number is palindrome or not. Input : 528 Output : It is not palindrome number Input : 545 Output : It is a palindrome number			
4.	Write an application which takes user input and displays the patterns as given below. (a) 1 (b)* (c)* 12 * * * 123 * * * * 1234 * * * * *			
5.	Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below: (Suppose we enter 3 names then the output should be as follows).. Number of arguments = 3 1 : First Student Name is = Arun 2 : Second Student Name is = Hiren 3 : Third Student Name is = Hitesh (Hint: Initialize string array with “First”, “Second”, etc.			
6.	Write a java application and make a class to show how Object-oriented programming is done. (a)Cube demo1 (b)Cube demo2 (c)Cube demo3 (d)Cube demo4 (e)Cube demo5			
7.	Create a student class with member roll no, name, birth-date and create multiple constructor.			
8.	Create a class “Employee” that would contain name and id as instance variables and count as a static variable. Define constructors to initialize variables of objects. Define methods to display variables’ value of objects which are created. The number of objects are maintained using count variable.			

9.	Write a Java application which takes input of date of birth and then calculates the age of the student accordingly.			
10.	Write a Java application to count and display frequency of letters and digits from the String given by the user as command-line argument.			
11.	Create a class “Rectangle” that would contain length and width as an instance variable and count as a static variable. Define constructors [constructor overloading (default, parameterized and copy)] to initialize variables of objects. Define methods to find area and to display variables’ value of objects which are created. [NOTE: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block]			
12.	Create a class “Student” that would contain enrollment No, name, and gender and marks as instance variables and count as static variables which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables. Also demonstrate constructor chaining. Create objects of class “Student” and display all values of objects.			
13.	Write a java program that implements method overloading on rectangle class by creating overloaded method area. Area is calculated using length and breadth in 1 method and using two points (x1,y1) and (x2,y2) in the other overloaded method.			

Name : Jain Apurva Sanjay
Roll No : 12
Subject : OOPJ

Department : M .C. A (Semester - 1)
Division : A
Subject Code : (MTCA13104

1. Install JDK and Write a Program to print Hello World on the console.

Code:-

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

Output:-

```
Hello World
```

2. Write an application which takes input from the command line and converts it into integer using parseInt().

Code:-

```
class Convert
{
        public static void main(String[ ]args)
        {
                int a;
                a = Integer.parseInt(args[0]);
                System.out.println(" your number * 10 = "+a * 10);
        }
}
```

Input:-

```
javac Convert.java
java Convert 2
```

Output:-

```
your number * 10 = 20
```

3. Write a java program to check whether a number is palindrome or not.
Input: 528 Output: It is not a palindrome number.
Input: 545 Output: It is not a palindrome number.

Code:-

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

```
	{
		int i,r,sum=0,temp,n=528;
		temp=n;
		for(i=0; n>0; i++)
		{
			r=n%10;
			sum=(sum*10)+r;
			n=n/10;
		}
		if(temp==sum)
		{
			System.out.println("palindrome number ");
		}
		else
		{
			System.out.println("not palindrome");
		}
	}
}
```

Output:-

```
not palindrome number
```

Code:-

```
class Palindrome
{
	public static void main(String args[])
	{
		int i,r,sum=0,temp,n=545;
		temp=n;
		for(i=0; n>0; i++)
		{
			r=n%10;
			sum=(sum*10)+r;
			n=n/10;
		}
		if(temp==sum)
		{
			System.out.println("palindrome number ");
		}
		else
		{
			System.out.println("not palindrome");
		}
	}
}
```

Output:-

```
palindrome number
```

4. Write an application which takes user input and displays the patterns as given below.

(a)
1
12
123
1234

(b)
*
*
*
*

(c)
*
* *
* * *
* * * *

a) **Code:-**

```
class NumTriangle
{
	public static void main(String[] args)
	{
		int size = Integer.parseInt(args[0]);
	for (int i = 1; i <= size; i++)
		{
		for (int j = 1; j <= i; j++)
			{
			System.out.print(j + " ");
		}
		System.out.println();
		}
	}
}
```

Input:-

```
java NumTriangle 4
```

Output:-

```
1
1 2
1 2 3
1 2 3 4
```

b) Code:-

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

Output:-

```
*
*
*
*
```

b) **Code:-**

```
class Triangle
{
        public static void main(String []args)
        {
                int n =5;
      for(int i =0 ; i<n;i++)
      {
       for(int j=0;j<=i;j++)
                {
                        System.out.print("*");
                }
        System.out.println("");
         }
        }
}
```

Output:-

```
*
**
***
****
*****
```

5. Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below: (Suppose we enter 3 names then output should be as follows).. Number of arguments = 3 1.: First Student Name is = Arun 2.: Second Student Name is = Hiren 3.Third Student Name is = Hitesh (Hint: Initialize string array with “First”, “Second”, etc.

Code:-

```
class CommandLineArguments
{
        public static void main(String[] args)
        {
                System.out.println("First Student Name  = "+args[0]);
                System.out.println("Second Student Name = "+args[1]);
                System.out.println("Third Student Name = "+args[2]);
                System.out.println("Fourth Student Name = "+args[3]);
        }
}
```

Input:-

```
java CommandLineArguments Apurva Sakshi Anand Dharmit
```

Output:-

```
First Student Name  = Apurva
Second Student Name = Sakshi
Third Student Name = Anand
Fourth Student Name = Dharmit
```

6. Write a java application and make a class to show how Object-oriented programming is done.
(a)Cube demo1 (b)Cube demo2 (c)Cube demo3 (d)Cube demo4
(e)Cube demo5

a) **Code:-**

```
class CubeDemo1
{
	public static void main(String[] args)
	{
		int len=97;
		int wid=71;
		int height=5;
		int volume = len*wid*height;
		System.out.println("Volume of cube = " + volume);
		if (volume>50)
		{
			System.out.println("Large cube");
		} else {
			System.out.println("Small cube");
		}
	}
}
```

Output:-

```
Volume of cube = 34435
Large cube
```

b) Code:-

```
class CubeDemo2
{
	public static void main(String[] args)
	{
		Cube2 c = new Cube2();
		c.len=71;
		c.wid=97;
		c.height=7;
		int volume = c.len*c.wid*c.height;
		System.out.println("Volume of cube = " +volume+"\t litres");
		if (volume>1000)
		{
			System.out.println("Large cube");
```

```
                } else {
                        System.out.println("Small cube");
                }
        }
}
class Cube2
{
        int len;
        int wid;
        int height;
}
```

Output:-

```
Volume of cube = 48209   litres
Large cube
```

c) Code:-

```
class CubeDemo3
{
        public static void main(String[] args)
        {
                Cube3 c = new Cube3();
                c.len=977;
                c.wid=71;
                c.height=11;
                c.volume();

        }
}
class Cube3
{
        int len;
        int wid;
        int height;
        void volume()
        {
                System.out.println("Volume of cube = " + (len*wid*height));
        }
}
```

Output:-

```
Volume of cube = 763037
```

d) Code:-

```
class CubeDemo4
{
        public static void main(String[] args)
        {
                Cube4 c = new Cube4();
                c.len=97;
```

```
                c.wid=71;
                c.height=13;
                System.out.println("Volume of cube = " + c.volume());
                if (c.volume()>10000)
                {
                        System.out.println("Large cube");
                } else {
                        System.out.println("Small cube");
                }
        }
}

class Cube4
{
        int len;
        int wid;
        int height;
        int volume()
        {
                 return len*wid*height;
        }
}
```

Output:-

```
Volume of cube = 89531
Large cube
```

e) Code:-

```
class CubeDemo5
{
        public static void main(String[] args) {
                Cube5 c = new Cube5(101, 51, 23);
                Cube5 c1 = new Cube5(7, 11, 13);
                System.out.println("Volume = " + c.volume());
                System.out.println("c1 Volume = " + c1.volume());
                if (c.volume() > 50)
                {
                        System.out.println("Large cube");
                } else {
                        System.out.println("Small cube");
                }

        }
}
class Cube5
{
        int len;
        int wid;
        int height;
        public Cube5(int l, int w, int height)
        {
                // Cube5 c = new Cube5(x, y, z);
```

```
                this.len = l;
                wid = w;
                this.height = height;
        }
        int volume()
        {
                return len*wid*height;
        }
}
```

Output:-

```
Volume = 118473
c1 Volume = 1001
Large cube
```

7. Create a student class with member roll no, name, birth-date and create multiple constructor.

Code:-

```
public class MultipleConstructor
{
        public static void main(String[] args)
        {
        Student s1 = new Student("Apurva", 12, "24/03/2003");
        Student s2 = new Student("Sakshi", 13, “12/11/2002");
                Student s3 = new Student("SaiPrasad", 144, "25/08/2000");

              s1.display();
              s2.display();
                 s3.display();
        }
}

class Student
{
        private String name;
        private int rollNo;
        private String birthDate;

        public Student(String name, int rollNo, String birthDate)
        {
        this.name = name;
        this.rollNo = rollNo;
        this.birthDate = birthDate;
        }

        public void display()
        {
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Birth Date: " + birthDate);
        System.out.println();
   }
```

```
}
```

Output:-

```
Name: Apurva
Roll No: 12
Birth Date: 24/03/2003

Name: Sakshi
Roll No: 13
Birth Date: 12/11/2002

Name: SaiPrasad
Roll No: 144
Birth Date: 25/08/2000
```

8. Create a class "Employee" that would contain name and id as instance variables and count as a static variable. Define constructors to initialize variables of objects. Define methods to display variables' value of objects which are created. The number of objects are maintained using count variable.

Code:-

```
public class Employee
{
        private String name;
        private int id;
        private static int count = 0;  // Static variable to maintain the count of objects

        // Constructor to initialize name and id
        public Employee(String name, int id)
        {
                this.name = name;
        this.id = id;
        count++;  // Increment the count when an object is created
        }

        // Method to display the details of the employee
        public void displayDetails()
        {
        System.out.println("Employee ID: " + id);
        System.out.println("Employee Name: " + name);
        }

   // Static method to display the count of objects created
        public static void displayCount()
        {
        System.out.println("Total Employee Count: " + count);
        }

        public static void main(String[] args)
        {
        // Create employee objects and display their details
        Employee employee1 = new Employee("Apurva", 4000);
        Employee employee2 = new Employee("Sakshi", 4200);
```

```
            System.out.println("Details of Employee 1:");
        employee1.displayDetails();

        System.out.println("\nDetails of Employee 2:");
        employee2.displayDetails();

        // Display the total count of employees
        Employee.displayCount();
        }
}
```

Output:-

Details of Employee 1:
Employee ID: 4000
Employee Name: Apurva

Details of Employee 2:
Employee ID: 4200
Employee Name: Krishna
Total Employee Count: Sakshi

9. Write a Java application which takes input of date of birth and then calculates the age of the student accordingly.

Code:-

```
import java.time.LocalDate;
import java.time.Period;

public class AgeGenerator
{
        public static void main(String[] args)
        {
    // get the input from the user
    String userInput = args[0];

    // parse the date from the input string
    LocalDate birthDate = LocalDate.parse(userInput);

    // calculate the age of the student
    LocalDate today = LocalDate.now();
    Period age = Period.between(birthDate, today);

    // print the age of the student
    System.out.println("The age of the student is: " + age.getYears() + " years.");
  }
}
```

Input:-

```
java AgeGenerator 2000-24-03
```

Output:-

```
The age of the student is: 21 years.
```

10. Write a Java application to count and display frequency of letters and digits from the String given by the user as a command-line argument.

Code:-

```
public class Frequency
{
	public static void main(String[] args)
	{
	String inputString = args[0];

		// Creating an array to store frequency of digits and letters
	int[] freqArray = new int[128];

		// Counting frequency of digits and letters
	for (int i = 0; i < inputString.length(); i++)
		{
		char c = inputString.charAt(i);
		if (Character.isLetterOrDigit(c))
			{
			freqArray[c]++;
		}
	}
		// Displaying frequency of digits and letters
		for (int i = 0; i < 128; i++)
		{
		if (freqArray[i] > 0)
			{
				System.out.println((char) i + ": " + freqArray[i]);
		}
	}
	}
}
```

Input:-

```
java Frequency Apurva
```

Output:-

```
java Frequency Apurva
a: 3
d: 1
i: 1
p: 1
r: 1
```

```
s: 2
```

11. Create a class “Rectangle” that would contain length and width as an instance variable and count as a static variable. Define constructors [constructor overloading (default, parameterized and copy)] to initialize variables of objects. Define methods to find area and to display variables’ value of objects which are created.

[NOTE: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block]

Code(default):-

```
class Bike1
{

	public static void main(String args[])
	{

		Bike b=new Bike();

	}
}

class Bike
{
	//creating a default constructor
	Bike()
	{
		System.out.println("Bike is created");
	}
}
```

Output:-

```
Bike is created
```

Code(paramiterized):-

```
class ParameterizedConstructor
{
	public static void main(String args[])
	{
		Student4 s1 = new Student4(12,"Apurva");
		Student4 s2 = new Student4(13,"Sakshi");
		s1.display();
		s2.display();
	}
}
class Student4
{
   int id;
   String name;
   Student4(int i,String n)
```

```
   {
        id = i;
        name = n;
   }

   void display()
   {
        System.out.println(id+" "+name);
   }
}
```

Output:-

```
12 Apurva
13 Sakshi
```

Code(copy):-

```
class CopyConstructor
{
        public static void main(String args[])
        {
                Student s1 = new Student(12,"Apurva");
                Student s2 = new Student(s1);
                s1.display();
                s2.display();
        }
}
class Student
{
   int id;
   String name;
   Student(int i,String n)
   {
        id = i;
         name = n;
   }

  Student(Student s)
  {
        id = s.id;
        name =s.name;
  }
   void display()
        {
                System.out.println(id+" "+name);
        }
}
```

Output:-

```
12 Apurva
12 Apurva
```

12. Create a class "Student" that would contain enrollment No, name, and gender and marks as instance variables and count as static variables which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables. Also demonstrate constructor chaining. Create objects of class "Student" and display all values of objects.

Code:-

```
class Student
{
	public static void main(String[] args)
	{
	Student s1 = new Student(1,  "Apurva", "Male", 85);
	Student s2 = new Student(2, "Sakshi", "Female");
	Student s3 = new Student(3, "Dharmit", "Male");

		s1.display();
	System.out.println();
	s2.display();
	System.out.println();
	s3.display();
	System.out.println();

		System.out.println("Total number of students: " + count);
	}


   static int count = 0;

   int rollNo;
   String name;
   String gender;
   float marks;

	Student(int rollNo, String name, String gender, float marks)
	{
	this.rollNo = rollNo;
	this.name = name;
	this.gender = gender;
	this.marks = marks;
	count++;
	}

	Student(int rollNo, String name, String gender)
	{
	this(rollNo, name, gender, 0);
	}

	void display()
	{
	System.out.println("Roll No: " + rollNo);
	System.out.println("Name: " + name);
	System.out.println("Gender: " + gender);
	System.out.println("Marks: " + marks);
	}
}
```

Output:-

```
Roll No: 1
Name: Apurva
Gender: Male
Marks: 95.0

Roll No: 2
Name: Sakshi
Gender: Female
Marks: 85.0

Roll No: 3
Name: Dharmit
Gender: Male
Marks: 0.0

Total number of students: 3
```

13. Write a java program that implements method overloading on rectangle class by creating an overloaded method area. Area is calculated using length and breadth in 1 method and using two points (x1,y1) and (x2,y2) in the other overloaded method.

Code:-

```
public class OverLoadingArea
{
   public static void main (String [] args)
   {
      //creating objects of rectangle
      Rectangle rectangle1 = new Rectangle(20,60);
      Rectangle rectangle2 = new Rectangle(10,20,20,40);

      //calling area method
      rectangle1.area();
      rectangle2.area();

      //calling perimeter method
      rectangle1.perimeter();
      rectangle2.perimeter();
   }
}

class Rectangle
{
   int length, breadth, x1, x2, y1, y2;

   Rectangle(int l, int b)
   {
      length = l;
      breadth = b;
   }

   Rectangle(int xl, int x2, int y1, int y2)
   {
      this.x1 = xl;
```

```
        this.x2 = x2;
        this.y1 = y1;
        this.y2 = y2;
    }

    void area()
    {
        int area = 0;

        if(length != 0 && breadth != 0)
        {
            area = length * breadth;
        }
        else
        {
            area = (x2 - x1) * (y2 - y1);
        }

        System.out.println("Area of the Rectangle: " + area);
    }

    void perimeter()
    {
        int perimeter = 0;

        if(length != 0 && breadth != 0)
        {
            perimeter = 2 * (length + breadth);
        }
        else
        {
            perimeter = 2 * ((x2 - x1)+(y2 -y1));
        }

        System.out.println("Perimeter of the Rectangle: " + perimeter);
    }
}
```

Output:-

```
Area of the Rectangle: 600
Area of the Rectangle: 200
Perimeter of the Rectangle: 100
Perimeter of the Rectangle: 60
```