CS3381 OBJECT ORIENTED PROGRAMMING LABORATORY

LIST OF EXPERIMENTS

NAME: **REG.NO:**

EXP. NO.	DATE	NAME OF THE EXPERIMENT	MARKS	SIGN
1		A.Sequential search		
		B.Binary search		
		C. Sorting algorithms-Selection Sort		
		D.Quadratic sorting algorithm-Insertion Sort		
2		A. Stack		
		B. Queue		
3		Generating pay slips for the employees		
4		Finding Area Of Different Shapes Using Abstract Class		
5		Finding Area Of Different Shapes Using Interface		
6		User defined exception handling.		
7		Multi-threaded application		
8		File Handling		
9		Finding Maximum and Minimum using Generic Classes		
10		Applications using JavaFX controls, layouts and menus.		
11		Mini project using Java concepts		

Ex. No. 5 **FINDING AREA OF DIFFERENT SHAPES USING INTERFACE**

Date:

Problem:

Write a Java Program to create an interface named Shape that contains an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes implements the class Shape. Each one of the classes contains variables and the method print Area () that prints the area of the given shape.

Aim:

To write a java program for finding area of different shapes using interface concept.

Algorithm:

Step 1: Start.
Step 2: Create an interface named Shape that contains an empty method named print Area()
Step 3: Get the value of length and breadth of rectangle.
Step 4: Calculate the area of rectangle using area=length * breadth
Step 5: Print the area of rectangle.
Step 6: Get the value of length and height of triangle.
Step 7: Calculate the area of triangle using area = 0.5 * length * height
Step 8: Print the area of circle.
Step 9: Get the value of radius.
Step 10: Calculate the area of circle using area =Math.PI * radius * radius.
Step 11: Print the area of circle.
Step 12: Stop.

Program:

```
import java.util.*;
interface Shape
{
        void printArea();
}
class Rectangle implements Shape

{
        int area, length,height;
        public void printArea()
        {
        System.out.println("\n   Rectangle\n ---------- ");
        Scanner input = new Scanner(System.in);
        System.out.printf("Enter Length of Rectangle : ");
        this.length = input.nextInt();
        System.out.printf("Enter Breadth of Rectangle : ");
        this.height = input.nextInt();
        this.area = this.length*this.height;
        System.out.println("Area of the Rectangle is : " + this.area);
        }
}
class Triangle implements Shape
{
        double area = 0.0;
        int length,height ;
        public void printArea()
        {
        System.out.println("\n   Triangle\n --------- ");
        Scanner input = new Scanner(System.in);
        System.out.printf("Enter Length of Triangle : ");
        this.length = input.nextInt();
        System.out.printf("Enter Height of Triangle : ");
        this.height = input.nextInt();
        this.area = 0.5 * this.length * this.height;
        System.out.println("Area of the Triangle is : " + this.area);
        }
}
class Circle implements Shape
{
        double area = 0.0;
        int length;
         public void printArea()
        {
        System.out.println("\n   Circle\n--------");
        Scanner input = new Scanner(System.in);
        System.out.printf("Enter Radius of Circle : ");
        this.length = input.nextInt();
        this.area = Math.PI * this.length * this.length;
        System.out.println("Area of the Circle is : "+this.area);
```

```
        }
}
class Main
{
public static void main(String[] args)
{
        System.out.println("\n----------\nFinding Area\n ---------- ");
        Shape rt = new Rectangle();
        rt.printArea();
        Shape tr = new Triangle();
        tr.printArea();
        Shape cr = new Circle();
        cr.printArea();
}
}
```

Viva Questions:

1.What is an interface in java?
2. Can an interface extends another interface?
3. How will you make classes Rectangle, Triangle, Circle to implement interface?
4. How will you get the input during runtime?
5. What modifiers are allowed for methods in an interface?

Result:

Thus the JAVA program for finding area of different shapes using interface concept was written, executed and the output was verified successfully.

Ex. No. 9 **FINDING MAXIMUM AND MINIMUM USING GENERIC CLASSES**

Date:

Problem:

Develop applications to demonstrate the features of generic classes.

Aim:

To write a java program for finding the maximum and minimum value from the given type of elements using a generic class.

Algorithm:

Step 1: Start.
Step 2: Create a class Myclass to implement generic class and generic methods.
Step 3: Get the set of values belonging to specific data type.
Step 4: Create the objects of the class to hold integer,character and double values.
Step 5: Create the methods to compare the values and find the maximum value and minimum value stored in the array.
Step 6: Invoke the methods with integer, character or double values.
Step 7: The maximum and minimum value of Integer ,character and double is displayed based on the data type passed to the method.
Step 8: Stop.

Program:

```
class MyClass<T extends Comparable<T>>
{
   T[] vals;
   MyClass(T[] obj)
   {
      vals = obj;
   }
   public T min()
   {
      T v = vals[0];
      for(int i=1; i < vals.length; i++)
      if(vals[i].compareTo(v) < 0)
      v = vals[i];
      return v;
   }
   public T max()
   {
      T v = vals[0];
      for(int i=1; i < vals.length;i++)
      if(vals[i].compareTo(v) > 0)
      v = vals[i];
      return v;
   }
}
```

```
class gendemo
{
   public static void main(String args[])
   {
      int i;
      Integer inums[]={10,2,5,4,6,1};
      Character chs[]={'j','p','x','a','n','d'};
      Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};
      MyClass<Integer> iob = new MyClass<Integer>(inums);
      MyClass<Character> cob = new MyClass<Character>(chs);
      MyClass<Double>dob = new MyClass<Double>(d);
      System.out.println("Max value in inums: " + iob.max());
      System.out.println("Min value in inums: " + iob.min());
      System.out.println("Max value in chs: " + cob.max());
      System.out.println("Min value in chs: " + cob.min());
      System.out.println("Max value in d: " + dob.max());
      System.out.println("Min value in d: " + dob.min());
   }
}
```

Viva Questions:

1.What is generics in java?

2.What are the advantages of using generics?

3.How do you declare a generic class?

4.What are generic methods?

5.What is type Erasure?

Result:

Thus the java program for finding the maximum and minimum value from the given type of elements was implemented using a generic class and executed successfully.

Ex. No. 11 **JAVAFX CONTROLS, LAYOUTS AND MENUS**
Date:

Problem:

Develop applications using JavaFX controls, layouts and menus.

Aim:

To write a java program to develop applications using JavaFX controls, layouts and menus.

Algorithm:

Step 1: Start
Step 2: Create a menu Button named “Technology”.
Step 3: Create necessary menu items and add it add it to the menu.
Step 4: Add the choice box to the scene.
Step 5: Set the stage and launch.
Step 6: Stop

Program:

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.MenuButton;
import javafx.scene.control.MenuItem;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
public class MenuButtonExample extends Application
{
  public void start(Stage stage)
  {
    MenuButton menu = new MenuButton("Technology");
    menu.setMnemonicParsing(true);
    MenuItem item1 = new MenuItem("Java");
    MenuItem item2 = new MenuItem("Python");
    MenuItem item3 = new MenuItem("C++");
    MenuItem item4 = new MenuItem("Big Data");
    MenuItem item5 = new MenuItem("Machine Learning");
    menu.getItems().addAll(item1, item2, item3, item4, item5);
    HBox layout = new HBox(25);
    layout.getChildren().addAll(menu);
    layout.setPadding(new Insets(15, 50, 50, 150));
    layout.setStyle("-fx-background-color: BEIGE");
    Scene scene = new Scene(layout, 595, 200);
    stage.setTitle("Menu Button");
    stage.setScene(scene);
    stage.show();
  }
```

```
    public static void main(String args[])
    {
            launch(args);
     }
}
```

Viva Questios:

1.What is javaFX?
2. What are various features of JavaFX?
3. Which method is used to create JavaFX application?
4. What is a Stage in JavaFX?
5. What is context menu in JavaFX?

Result:

Thus the application using JavaFX controls, layouts and menus was developed and the output was verified successfully.