| **Problem Statement : Define a application to find the area() and perimeter() of various shapes** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| **What?** | | | |  | **How?** | | | |
| 1. What are the shapes we have to consider?  **Ans: Circle, Rectangle, Square and Triangle.**  2. What are the parametes we have to consider?  **Ans: Length, breadth, side, radius, height**  3. What are the formulas we have to use to calculate area and  perimeter?  **Ans: Area- Circle: pi\*r^2, Rectangle: length\*breadth,**  **Square: side\*side, Triangle: 1/2\*breadth\*height.**  **Perimeter:** **Circle: 2\*pi\*radius,**  **Rectangle: 2\*(length+breadth),Square: 4\*side,**  **Triangle: side1+side2+side3.**  4. Any predefined values are required?  **Ans: pi = 3.14** | | | |  | 1.Using Single class read all the required inputs and  calculate area and Perimeter, And display the result.  2.Using Single class and method overloading  calculate area and perimeter,And display the result.  3.Using different classes for various shapes and  calculate area and perimeter And displaythe result.  **4.Using different classes for various shapes and**  **inherit the common properties from a class called**  **"Shape" and calculate area and perimeter and display the result.**  **5.Using different classes for various shapes and**  **inherit the common properties from a abstract class called**  **"Shape" and calculate area and perimeter and display the result.**  **6. Using different classes for various shapes and**  **inherit the common properties from a class called**  **"Shape" and implements an interface called "ShapePlan" and**  **calculate area and perimeter and display the result.** | | | |
|
|
|
|
|
|
|
|
|
|
|
|
|  |
|  |
|  |
|  |
|  | | | | | | | | |
| Using different classes for various shapes and  inherit the common properties from a class called  "Shape" and implements an interface called "ShapePlan" and  calculate area and perimeter and display the result.  We can only declare in interface (Secure).  We can declare and also can define in abstract.  We can achieve 100% abstraction in interface.  Code reusability.  We can make a plan by using interface.  Code flexibility (Updation). | | | |  | Compare than this two, that is more secure.  This two comes without interface.  We can't achieve 100% abstraction.  The second one is partially completed.  When we compare to that code flexibility is less efficient  in this two.  We can't plan in this two solutions because of absence  of interface. | | | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Why?** | | | |  | **Why Not?** | | | |

**Algorithm**

**Step 1:** Start

**Step 2:** Get the input from the user.(Shapes and parameters).

**Step 3:** Make an interface class named ShapePlan.

**Step 4:** Define a method perimeter and area and a variable pi = 3.14 in the interface.

**Step 5:** Make an abstract class named Shapes and implement ShapePlan.

**Step 6:** Inherit the abstract class and override the methods.

**Step 7:** Declare variables(length, breadth, side, radius).

**Step 8:** Give the corresponding formulas inside the methods.

**Step 9:** Stop.

**Coding**

**interface ShapePlan {**

**void area();**

**void perimeter();**

**double *pi* = 3.14;**

**}**

**abstract class Shapes implements ShapePlan {**

**public abstract void area();**

**public abstract void perimeter();**

**}**

**class Circle extends Shapes {**

**float radius;**

**Circle(float radius){**

**this.radius= radius;**

**}**

**public void area() {**

**System.*out*.println("Area of Circle: "+*pi*\*radius\*radius);**

**}**

**public void perimeter() {**

**System.*out*.println("Perimeter of Circle: "+(2\**pi*\*radius));**

**}**

**}**

**class Rectangle extends Shapes {**

**int length, breadth;**

**Rectangle(int length, int breadth){**

**this.length= length;**

**this.breadth = breadth;**

**}**

**public void area() {**

**System.*out*.println("Area of Rectangle: "+length\*breadth);**

**}**

**public void perimeter() {**

**System.*out*.println("Perimeter of Rectangle: "+ (2\*(length+breadth)));**

**}**

**}**

**class Square extends Shapes {**

**int side;**

**Square(int side){**

**this.side = side;**

**}**

**public void area() {**

**System.*out*.println("Area of Square: "+ side\*side);**

**}**

**public void perimeter() {**

**System.*out*.println("Perimeter of Square: "+4\*side);**

**}**

**}**

**class Triangle extends Shapes {**

**int height, breadth, side;**

**Triangle(int breadth, int height, int side){**

**this.breadth = breadth;**

**this.height = height;**

**this.side = side;**

**}**

**public void area() {**

**System.*out*.println("Area of Triangle: "+ (0.5 \* breadth \* height));**

**}**

**public void perimeter() {**

**System.*out*.println("Perimeter of Triangle: "+ (breadth+height+side));**

**}**

**}**

**public class ShapesPrblmSolving {**

**public static void main(String[] args) {**

**System.*out*.println("\n\t\*\*\*\*\*Circle\*\*\*\*\*\n");**

**Shapes obj = new Circle(4);**

**obj.area();**

**obj.perimeter();**

**System.*out*.println("\n\t\*\*\*\*\*Rectangle\*\*\*\*\*\n");**

**Shapes obj1 = new Rectangle(15,45);**

**obj1.area();**

**obj1.perimeter();**

**System.*out*.println("\n\t\*\*\*\*\*Square\*\*\*\*\*\n");**

**Shapes obj2 = new Square(4);**

**obj2.area();**

**obj2.perimeter();**

**System.*out*.println("\n\t\*\*\*\*\*Triangle\*\*\*\*\*\n");**

**Shapes obj3 = new Triangle(4,6,8);**

**obj3.area();**

**obj3.perimeter();**

**}**

**}**