**Write an algorithm to find the area and perimeter of a various shapes (square,rectangle,circle,Triangle)**

**Algorithm:**

**Step-1**:Start.

**Step-2:**create an interface called “Shape plan”.

**Step-3:**Declare an area and perimeter as an abstract method inside the interface and declare the PI as constant .

**Step-4**:create a superclass as an abstract class called “Shapes” implements interface (ShapePlan).

**Step-5**  Declare length,breadth,height and side variables as instances inside abstract class .

**Step-6:**create constructor for initialize the values of length,breadth,height,side.

**Step-7:**Declare the methods area and perimeter,

**Step-8:**create classes called rectangle, circle, triangle and Squares extend shapes and define methods to find area and perimeter of all classes by applying the correct formula.

**Step-9:**create object for subclass and pass the values for length,breadth,height,side.

**Step-10:**Display the area and perimeter.

**Step-10.**Stop.

| **PROBLEM STATEMENT:Define an application to find area and perimeter o f various shapes**  **[ Ex:Square,Rectangle,Circle,Triangle ]** | | | | | | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| **What?** | | | |  | **How?** | | | |
| 1.What are the shapes we have to consider?  **Ans:**Circle, Rectangle,Square and Triangle  2)What are the parameters we have to consider?  **Ans:**Length,breadth,side,radius  3.What are the formula we have to use to  calculate area?  **Ans:Circle:**PI\*R^2,**Rectangle:**Length\*Breadth,  **Square**:Side \*Side **Triangle:**1/2\*Breadth\*Height  4)What are the formula we have to use to  calculate Perimeter?  **Ans:Circle:**2\*PI\*R,**Rectangle:**2\***(**Length+Breadth)  **Square**:4\*Side ,**Triangle:**side1+side2+side3  5Any predefined Values are required? | | | |  | 1)Using Single class read all the required value 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 display the result.  4)Using different classes for various shapes and inherit the common  propertiesfrom the class called "Shape".and calculate the area and perimeter  and display the result.  5)Using different classes for various shapes and inherit the common  properties from theabstract class called "Shape".and calculate the area and  perimeter and display the result.  6Using different classes for various shapes and inherit the common  properties from theabstract class called "Shape".and implements an interface  called "Shape"Plan calculate the area and perimeter and display the result. | | | |
|  |  |  |  |  |  |  |  |  |
| **Why?** | | | |  | **Why not ?** | | | |
| 6)Using different classes for various shapes and inherit the common  properties from the abstract class called "Shape".and implements an interface  called "Shape"Plan calculate the area and perimeter and display the result.  **Reason:**  1)Code reusability by using inheritance  2)Using interface we can achieve full abstraction(100%).  3)Interface is used to make an plan.  4)Using Interface we can obtain multiple inheritance  5)We can declare and also can define in abstract.  6)We can declare in the interface (secure). | | | |  | 4)Using different classes for various shapes and inherit the common  propertiesfrom the class called "Shape".and calculate the area and perimeter  and display the result.  5)Using different classes for various shapes and inherit the common  properties from theabstract class called "Shape".and calculate the area and  perimeter and display the result.  Reason:  1)we cannot create an object(instantiate) for abstract class.  2)Abstract class doesnt supprot multiple inheritane.  3) We can't able to reuse the code without inheritance.  4) We can't able to achieve 100% abstraction without interface.  5) Without inheritance we cant able to add a new functionality without  affecting the existing one.  6) We want to call each and every class for displaying the result.  7) We cant able to group the common classes. | | | |

Code:

**package** com.aravind.day23;

**interface** Shapeaplan {

**double** ***pi*** = 3.14;

**void** area();

**void** perimeter();

}

**abstract** **class** Shape **implements** Shapeaplan{

//constructor

**public** **abstract** **void** area();

**public** **abstract** **void** perimeter();

}

// subclass Square

**class** Square **extends** Shape {

**int** side;

Square(**int** side )

{

**this**.side =side;

}

**public** **void** area()

{

System.***out***.println(side\*side);

}

**public** **void** perimeter()

{

System.***out***.println(4\*side);

}

}

// Subclass circle

**class** Circle **extends** Shape

{

**int** radius;

Circle(**int** r )

{

**this**.radius=r;

}

**public** **void** area()

{

System.***out***.println(***pi***\*(radius\*radius));

}

**public** **void** perimeter()

{

System.***out***.println(2\****pi***\*radius);

}

}

**class** Rectangle **extends** Shape

{

**int** length, breadth;

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

{

**this**.length=length;

**this**.breadth=breadth;

}

**public** **void** area()

{

System.***out***.println(length\*breadth);

}

**public** **void** perimeter()

{

System.***out***.println(2\*(length+breadth));

}

}

**class** Triangle **extends** Shape{

**int** length;

**int** breadth;

**int** side1,side2,side3;

Triangle (**int** length,**int** breadth,**int** side1,**int** side2,**int** side3)

{

**this**.length=length;

**this**.breadth=breadth;

**this**.side1=side1;

**this**.side2=side2;

**this**.side3=side3;

}

**public** **void** area()

{

System.***out***.println(length\*breadth/2);

}

**public** **void** perimeter()

{

System.***out***.println(side1+side2+side3);

}

}

**public** **class** findAreaandPerimeter {

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

Square obj=**new** Square(2);

obj.area();

obj.perimeter();

Circle obj1=**new** Circle(2);

obj1.area();

obj1.perimeter();

Rectangle obj2=**new** Rectangle(4,4);

obj2.area();

obj2.perimeter();

Triangle obj3=**new** Triangle(2,2,1,2,3);

obj3.area();

obj3.perimeter();

}

}