

3. Problem Statement

1. Create an abstract class Figure with following properties and functions:

Properties: double dim1;

Methods: abstract void findArea();

abstract void findPerimeter();

Create three subclasses Circle, Rectangle and Triangle that extends Figure class and define both the methods.

Write a program that will find the area and perimeter of 3 Figures and print the details for all.

2. Declare an integer array of size 10. Initialize using for loop with 1 to 10, and print all even numbers from an array.

3. Write a program to generate a user-defined exception called NegativeAgeException if the user inputs negative value for age.

Solution

1. Create an abstract class Figure with following properties and functions:

Properties: double dim1;

Methods: abstract void findArea();

abstract void findPerimeter();

Create three subclasses Circle, Rectangle and Triangle that extends Figure class and define both the methods.

Write a program that will find the area and perimeter of 3 Figures and print the details for all.

```
import java.io.*;
```

```
abstract class Values// making one abstract class
```

```
{  
  
    //abstract functions  
  
    abstract public void input() throws IOException;  
  
    abstract public float area();  
  
    abstract public float perimeter();  
  
}
```

class Rectangle extends Values // inherits

```
{  
  
    int length;  
  
    int breath;  
  
    public void input()throws IOException  
    {  
  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
  
        System.out.println("Enter length");  
  
  
        length=Integer.parseInt(br.readLine());  
  
        System.out.println("Enter breath ");  
  
  
        breath=Integer.parseInt(br.readLine());  
    }  
  
    public float area()  
    {  
  
        int Area=length*breath;  
  
        return Area;  
    }  
  
  
    public float perimeter()  
    {
```

```

        int Perimeter= 2*(length+breath);
        return Perimeter;
    }
}

```

class Triangle extends Values

```

{
    int length;
    int base;
    int side;
    public void input()throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter all three sides of Triangle ");
        length=Integer.parseInt(br.readLine());
        base=Integer.parseInt(br.readLine());
        side=Integer.parseInt(br.readLine());
    }
    public float area()
    {
        float Area= 0.5f*(length*base);
        return Area;
    }
    public float perimeter()
    {
        int Perimeter= length+base+side;
        return Perimeter;
    }
}

```

class Circle extends Values

```
{  
    int radius;  
  
    public void input()throws IOException  
    {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        System.out.println("Enter radius ");  
        radius=Integer.parseInt(br.readLine());  
    }  
    public float area()  
    {  
        float Area=3.14f*radius*radius;  
        return Area;  
    }  
    public float perimeter()  
    {  
        float Perimeter= 3.14f*radius*2;  
        return Perimeter;  
    }  
}
```

class Geometry // main class

```
{  
    public static void main(String args[])throws IOException
```

```
{  
//polymorphism  
Values ref;// reference of Values class  
  
ref=new Rectangle(); // using ref to define the abstract functions with values and print them  
ref.input();  
System.out.println("Area of rectangle " +ref.area());  
System.out.println("perimeter of rectangle " +ref.perimeter());  
  
ref=new Triangle(); // using ref to define the abstract functions with values and print them  
ref.input();  
System.out.println("Area of Triangle " +ref.area());  
System.out.println("perimeter of Triangle " +ref.perimeter());  
  
ref=new Circle(); // using ref to define the abstract functions with values and print them  
ref.input();  
System.out.println("Area of Circle " +ref.area());  
System.out.println("Area of Circle " +ref.perimeter());  
  
}  
}
```

Screenshot for the result

```

        at Geomtery.main<Geomtery.java:93>
C:\sweety_backup\Training\Assignment4> javac Geomtery.java
C:\sweety_backup\Training\Assignment4> java Geomtery
Enter length
10
Enter breath
10
Area of rectangle 100.0
perimeter of rectangle 40.0
Enter all three sides of Triangle
12
12
12
Area of Triangle 72.0
perimeter of Triangle 36.0
Enter radius
14
Area of Circle 615.44006
Area of Circle 87.920006
C:\sweety_backup\Training\Assignment4>

```

2. Declare an integer array of size 10. Initialize using for loop with 1 to 10, and print all even numbers from an array.

```

import java.util.Scanner;

class Arraynum
{
    public static void main(String[] a)
    {
        Scanner s1= new Scanner(System.in);
        int num[] = new int[10];

        for(int i=0;i<10;i++)
        {
            System.out.println("Enter the number");
            num[i] =s1.nextInt();
        }
        for (int i = 0; i <10; i++)

```

```

{
if (num[i]%2==0)

System.out.println("the even numbers is "+num[i]);

}

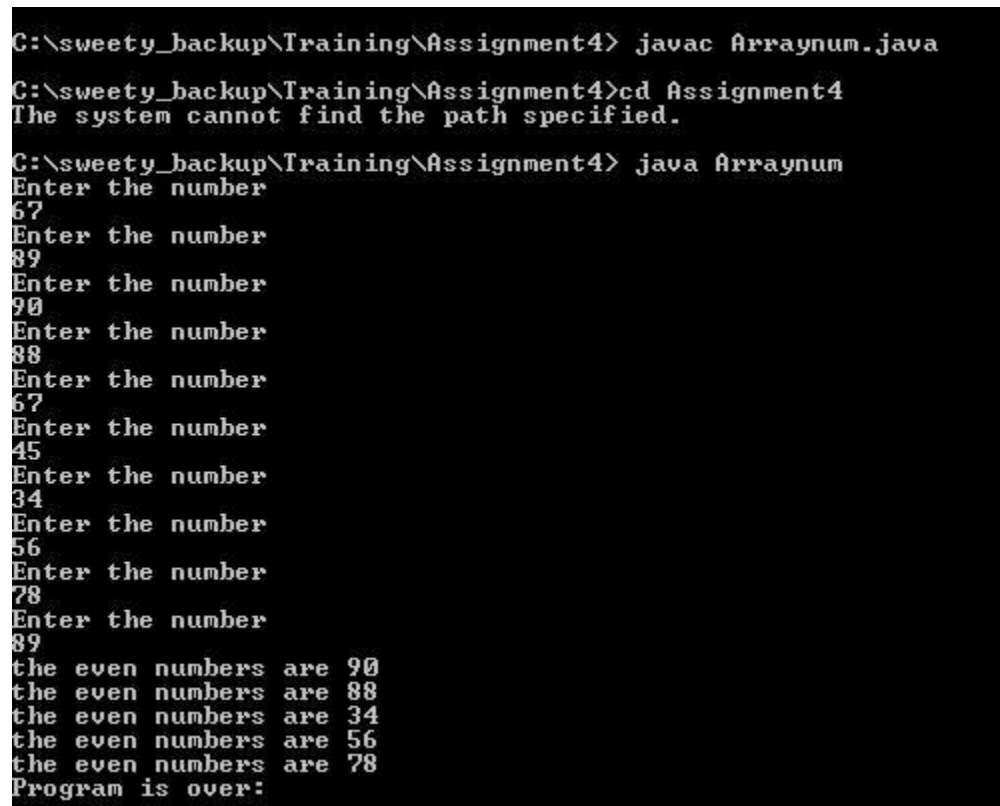
System.out.println("Program is over:");

}

}

```

ScreenShot for the result



```

C:\sweety_backup\Training\Assignment4> javac Arraynum.java
C:\sweety_backup\Training\Assignment4>cd Assignment4
The system cannot find the path specified.
C:\sweety_backup\Training\Assignment4> java Arraynum
Enter the number
67
Enter the number
89
Enter the number
90
Enter the number
88
Enter the number
67
Enter the number
45
Enter the number
34
Enter the number
56
Enter the number
78
Enter the number
89
the even numbers are 90
the even numbers are 88
the even numbers are 34
the even numbers are 56
the even numbers are 78
Program is over:

```

3. Write a program to generate a user-defined exception called NegativeAgeException if the user inputs negative value for age.

```

class AgeNegative extends Exception{

    AgeNegative(String s)

    {

        super(s);

    }

}

```

```
}
```

```
class NegativeAgeException
```

```
{
```

```
    static void validate(int age) throws AgeNegative // using validate function
```

```
    {
```

```
        if(age < 0)
```

```
        {
```

```
            throw new AgeNegative("Age cant be Negative");
```

```
        }
```

```
    else
```

```
        System.out.println("Age is fine");
```

```
    }
```

```
    public static void main(String[] args)
```

```
    {
```

```
        try
```

```
        {
```

```
            validate(56); // defining value
```

```
            validate(-90);
```

```
            validate(0);
```

```
        }
```

```
        catch (Exception a)
```

```
        {
```

```
            System.out.println("Age Invalid Eception Occured");
```

```
        }
```

```
        System.out.println("End.....");
```

```
    }
```

```
}
```


Screensot for the Result

```
C:\sweety_backup\Training\Assignment4> javac NegativeAgeException.java
C:\sweety_backup\Training\Assignment4> java NegativeAgeException
Age is fine
Age Invalid Eception Occured
End.....
```