Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**9**
Student Name:	**Shreyas Sanjay Nanaware**
Roll No :	**39**

Title:

1. Write a java program to create an abstract class named Shape that contains two integers and an abstract method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

Learning Objective:

Students will be able to implement abstract class and abstract method programs.

Learning Outcome:

- Understanding the abstraction concept and hiding of the unnecessary code.

Course Outcome:

ECL304.4	1. Implement different programming applications using packaging.

Faculty: Ms. Deepali Kayande

Theory:

- Explain in details about necessity of data hiding in any application / project.

Ans: 1.Data hiding ensures exclusive data access to class members and protects object integrity by preventing unintended or intended changes. Data hiding also reduces system complexity for increased robustness by limiting interdependencies between software components.

2. Data hiding can be very useful in cases where like for example there are multiple people working on a single project and each one of them represent different department. In this case we can take an school/college example,
In a college the one subject department should not have access to the work of the other department's and vice versa is also applicable here.

3. Data hiding is a method used in object-oriented programming to hide information within computer code. One advantage of data hiding is heightened security against hackers. Data hiding takes certain parts of code and hides those parts from the objects. The objects cannot directly access any data that is hidden.

4. At the same time, hiding data can make it harder for a programmer, who may need to use more code to create effects in hidden data than would be necessary if the data were public. But this scenario can be eliminated if from the start we declare what all things are needed to be public and what to be private to eliminate the confusion in the later stages of the program.

5. Object-oriented programming is a type of programming in which parts of the code are split into objects. Each of these objects is programmed in capsules, so each object has its own coding that applies only to that object. Without specifying data as hidden, all the data is open to the objects.

6. The advantage for programmers is that there is no way for a programmer to accidentally link to incorrect data. Data hiding ensures that, if a programmer does make this link, the program will simply return an error so the programmer can quickly correct the mistake. This also ensures that all of the objects are truly isolated units, which is the main concept of object-oriented coding. Volatile data are typically hidden because, if such data were made public, it could damage the object and destroy the entire program.

- **Explain abstract class and abstract methods.**

Ans: 1. **Abstract Class :** A class which contains the abstract keyword in its declaration is known as abstract class.

2.An abstract class can have both abstract methods and normal methods at the same time. Abstract methods are the one in which there is only declaration of the method name in the class or interface for that matter of fact and the actual method body is declared outside of the class.

3.Its object cannot be created or in other words it cannot be instantiated.

4. In an interface it is by default that whatever method we are declaring is always abstract and its body is always outside that of the class in which it has been declared.

5. **Abstract class:** is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class). Abstract method: can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from).

6. The purpose of an abstract class is to provide a blueprint for derived classes and set some rules what the derived classes must implement when they inherit an abstract class. We can use an abstract class as a base class and all derived classes must implement abstract definitions.

7. Java Abstract class is used to provide common method implementation to all the subclasses or to provide default implementation. We can run abstract class in java like any other class if it has main() method.

8. Similar to the interface we can define static methods in an abstract class that can be called independently without an object. Abstract classes can also have final methods (methods that cannot be overridden)

9.Following given below is the syntax for abstract class and method respectively:
Abstract class syntax: abstract class A{}
Abstract method syntax: abstract void printStatus() ; //no method body

1. **Write a java program to create an abstract class named Shape that contains two integers and an abstract method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.**

Algorithm:

Step 1: Create an abstract class named 'Shape' which contains 2 integers and an abstract method named printArea().

Step 2: Create 3 classes namely , Rectangle , Triangle and Circle. All the 3 inherited classes of Shape class (using hierarchical inheritance) has the same method printArea() with formulas for area of the respective shapes.

Step 3: For each of the shape take the input values form the user for the sides of the respective shape and then perform the calculation accordingly.

Step 4: In the main function, create the 3 objects of the abstract class representing the 3 subclasses and calling each of the 3 methods for displaying the output.
(although we now that the object of the abstract class cannot be created but as the objects are calling the methods of the subclasses, we will get the desired output)

Program:

```
/*Write a java program to create an abstract class named Shape that contains
two integers and an abstract method named printArea(). Provide three classes
named
Rectangle, Triangle and Circle such that each one of the classes extends
the class Shape. Each one of the classes contain only the method printArea()
that prints the area of the given shape.
*/
package javaprogramming3;

import java.util.Scanner;
```

```
abstract class Shape {   //creating an abstract class using the abstract keyowrd
        int num1 , num2;    //declaring the attributes of the class
        abstract void printArea();    //declaring the abstract method
        Scanner in= new Scanner(System.in);
}

class Rectangle extends Shape  //inheriting the parent class using the concept of
hierarchical inheritance
{
        void printArea() {
                System.out.println("Enter length of the rectangle: ");   //taking user
input and storing it
                num1=in.nextInt();
                System.out.println("Enter breadth of the rectangle: ");
                num2=in.nextInt();
                System.out.println("Area of Rectangle = Length X Breadth = " +
(num1*num2) + " sq. units \n"); //calculating the area and displaying the same

        }
}

class Triangle extends Shape   //inheriting the parent class using the concept of
hierarchical inheritance
{
        void printArea() {
                System.out.println("Enter the base of the triangle : ");    //taking
user input and storing it
                num1=in.nextInt();
                System.out.println("Enter the height of the triangle : ");
                num2=in.nextInt();
                System.out.println("Area of Triangle  = 1/2 X Base X Height = " +
(0.5*num1*num2) + " sq. units \n");   //calculating the area and displaying the
same
        }
}

class Circle extends Shape  //inheriting the parent class using the concept of
hierarchical inheritance
{
        void printArea() {
                System.out.println("Enter the radius of the circle : ");  //taking user input and storing it
                num1=in.nextInt();
                float area = (float) (3.141592*num1*num1);
                System.out.println("Area of Circle  = 3.14 X radius X radius = " + area + " sq. units \n");   //calculating the area and displaying the same
        }
}

public class Lab9 {
        public static void main(String[] args) {
                Shape rectangle = new Rectangle();  //creating the objects of the abstract class
                Shape triangle = new Triangle();
                Shape circle = new Circle();
                rectangle.printArea();  //calling the methods using the object
                triangle.printArea();
                circle.printArea();
        }
}
```

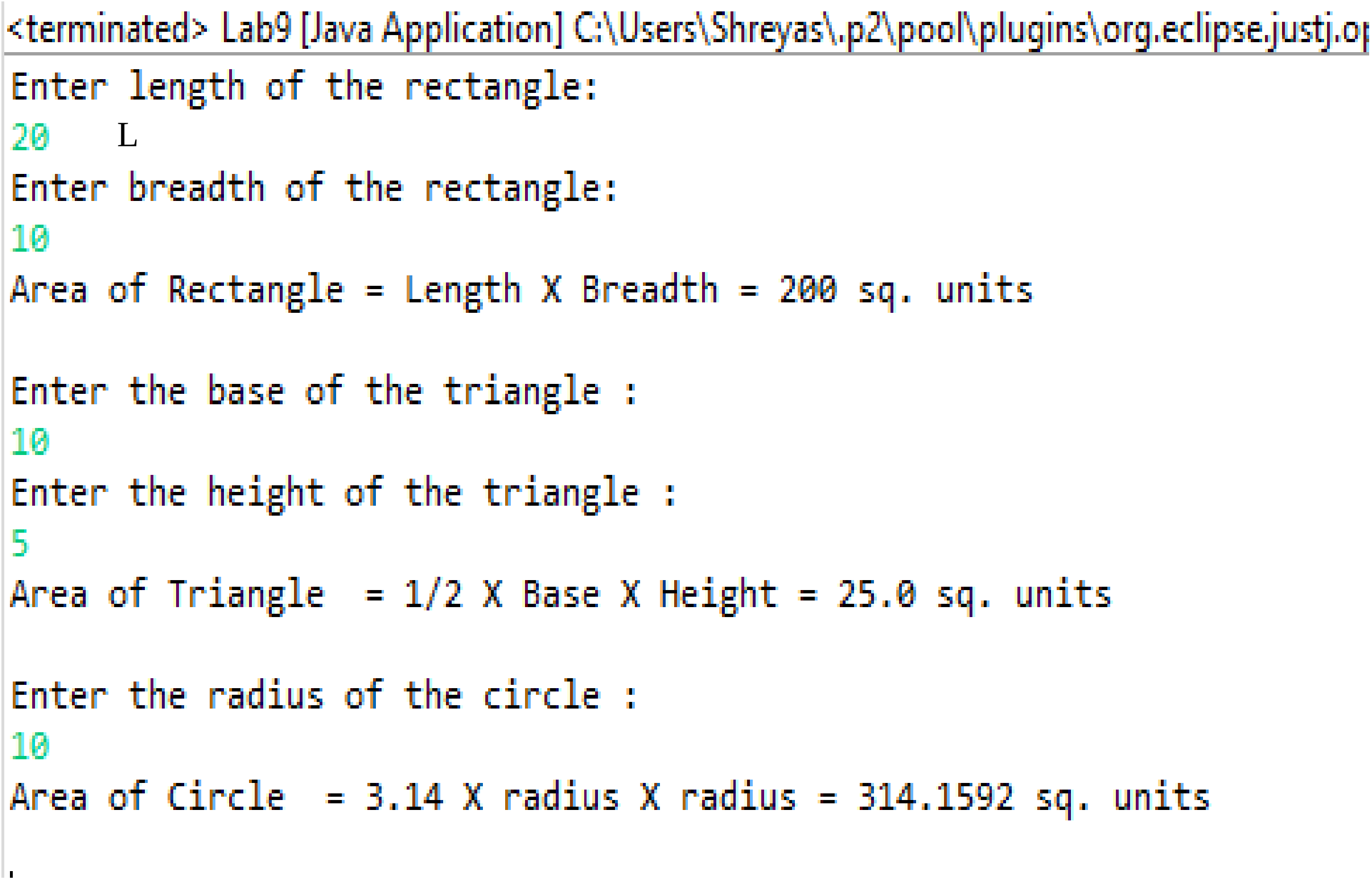
Input Given:

Length of rectangle: 20 , Breadth of rectangle: 10

Base of triangle: 10 , Height of triangle : 5

Radius of Circle: 10

Output:



```
<terminated> Lab9 [Java Application] C:\Users\Shreyas\.p2\pool\plugins\org.eclipse.justj.o
Enter length of the rectangle:
20    L
Enter breadth of the rectangle:
10
Area of Rectangle = Length X Breadth = 200 sq. units

Enter the base of the triangle :
10
Enter the height of the triangle :
5
Area of Triangle  = 1/2 X Base X Height = 25.0 sq. units

Enter the radius of the circle :
10
Area of Circle  = 3.14 X radius X radius = 314.1592 sq. units
```