Lab Number:	9
Student Name:	**Bhagyesh Subhash Manjrekar**
Roll No :	**07**

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:

- Implement different programming applications using packaging.

Faculty: Ms. Deepali Kayande

Theory:

(1) Explain in details about necessity of data hiding in any application / project .

Ans:
Data hiding is a technique of hiding internal object details, i.e., data members. It is an object-oriented programming technique. Data hiding ensures, or we can say guarantees to restrict the data access to class members. It maintains data integrity.

Data hiding means hiding the internal data within the class to prevent its direct access from outside the class. Data encapsulation hides the private methods and class data parts, whereas Data hiding only hides class data components. Both data hiding and data encapsulation are essential concepts of object-oriented programming. Encapsulation wraps up the complex data to present a simpler view to the user, whereas Data hiding restricts the data use to assure data security.

Data hiding also helps to reduce the system complexity to increase the robustness by limiting the interdependencies between software components. Data hiding is achieved by using the private access specifier.
Data Hiding in java helps you to protect your data from outside interference and provides data integrity logic. Data hiding also reduces system complexity for increased robustness by limiting interdependencies between software components.

Data hiding is also known as data encapsulation or information hiding.

(2) Explain abstract class and abstract methods.
Ans:
Data abstraction is the process of hiding certain details and showing only essential information to the user. Abstraction can be achieved with either abstract classes or interfaces.

The abstract keyword is a non-access modifier, used for classes and methods.

Abstract class:
It is a restricted class that cannot be used to create objects. A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract

Abstract class in Java is similar to interface except that it can contain default method implementation. An abstract class is mostly used to provide a base for subclasses to extend and implement the abstract methods and override or use the implemented methods in abstract class.

Abstract method: It can only be used in an abstract class, and it does not have a body. A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

An abstract method is a method that is declared without an implementation.

Faculty: Ms. Deepali Kayande

Statement: 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: Start

Step 2: Using the abstract keyword, abstract the class shape

Step 3: Create an abstract class named Shape that contains two integers and an abstract method named printArea().

Step 4: Create three classes named Rectangle, Triangle and Circle, and extend each class with shape class

Step 5: At last prints the area of the given shape

Step 6: Stop

Program: import java.util.Scanner;

abstract class shape

{

abstract void printarea();

}

class Rectangle extends shape

{

Scanner in=new Scanner(System.in);

int a1,l,b;

void printarea()

```java
{
System.out.println("Enter length:");

l=in.nextInt();

System.out.println("Enter Breadth:");

b=in.nextInt();

a1=l*b;

System.out.println("Area of Rectangle is: " +a1);

}

}

class Triangle extends shape

{

Scanner in=new Scanner(System.in);

double a2;

int ba,h;

void printarea()

{

System.out.println("Enter base:");

ba=in.nextInt();

System.out.println("Enter height:");

h=in.nextInt();

a2=0.5*ba*h;

System.out.println("Area of Triangle is: " +a2);

}

}

class Circle extends shape

{
```

```java
Scanner in=new Scanner(System.in);

double r,a3;

void printarea()

{

System.out.println("Enter radius:");

r=in.nextInt();

a3=3.14*r*r;

System.out.println("Area of Circle is: " +a3);

}

}

public class abstract_shape

{

public static void main(String[] args)

{
// TODO Auto-generated method stub

shape r=new Rectangle();

shape t=new Triangle();

shape c=new Circle();

r.printarea();

t.printarea();

c.printarea();}}
```
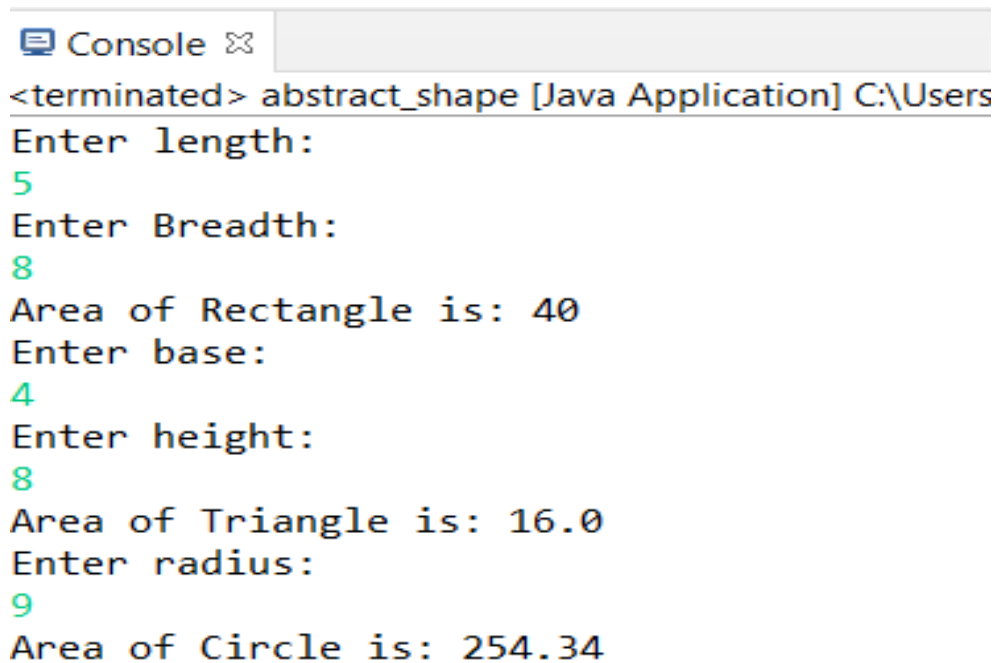
Input given:

a1=l*b;

a2=0.5*ba*h;

a3=3.14*r*r;

Output Screenshot:

```
Console ⌗
<terminated> abstract_shape [Java Application] C:\Users
Enter length:
5
Enter Breadth:
8
Area of Rectangle is: 40
Enter base:
4
Enter height:
8
Area of Triangle is: 16.0
Enter radius:
9
Area of Circle is: 254.34
```