Sukkur Institute of Business Administration University

Department of Computer Science

Object Oriented Programming using Java

BS – II (CS/AI/SE)

Spring-2024

Lab # 11: Let's learn about Abstraction (Interfaces and Abstract classes)

Instructor: Nimra Mughal

Objectives

After performing this lab, students will be able to understand:

- Interfaces
- Abstract classes
- Exception Handling

Interfaces vs Abstract Classes

Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

But there are many differences between abstract class and interface that are given below.

Abstract class	Interface
Abstract class can **have abstract and non- abstract** methods.	Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also.
Abstract class **doesn't support multiple inheritance**.	Interface **supports multiple inheritance**.
Abstract class **can have final, non-final, static and non-static variables**.	Interface has **only static and final variables**.
Abstract class **can provide the implementation of interface**.	Interface **can't provide the implementation of abstract class**.
The **abstract keyword** is used to declare abstract class.	The **interface keyword** is used to declare interface.
An **abstract class** can extend another Java class and implement multiple Java interfaces.	An **interface** can extend another Java interface only.
An **abstract class** can be extended using keyword "extends".	An **interface** can be implemented using keyword "implements".
A Java **abstract class** can have class members like private, protected, etc.	Members of a Java interface are public by default.

Example:
 	
public abstract class Shape{
 	
 public abstract void draw();
 	
}
	Example:
	
	public interface Drawable{
	
	   void draw();
	
	}

Links for practice:

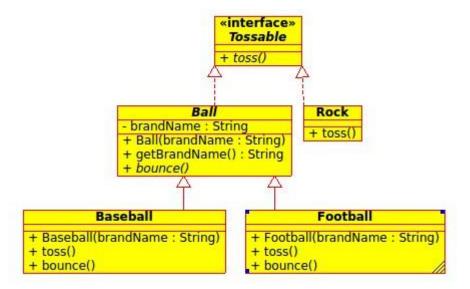[30 Java Practice Coding Questions On Abstract Classes (javaconceptoftheday.com)](javaconceptoftheday.com)

[Practice questions of Java - Java Abstract class (codesdope.com)](codesdope.com)

[Difference between Abstract class and Interface - Javatpoint](Javatpoint)

Exercises

Question: 1

Implement the following class hierarchy. You do not need to fill in the method bodies with actual code for the **toss** or **bounce** methods. You can use string message or whatever suitable you want.

Question: 2

Declare the Rectangle, SportCar, Manager classes as classes that implement the Printable interface and run the given application. The output of this application should be the details of each one of the objects that were instantiated.

```java
interface Printable
{
    public void print();
}
public class PrintableDemo
{
    public static void main(String args[])
    {   Printable vec[] =
{new Rectangle(110,80), new SportCar("Toyota", 989621),
                    new Rectangle(34,32), new Manager("John",
40),
                        new Rectangle(54,10), new SportCar("Audi",
2365644),
                            new SportCar("Mazda", 4322343), new Manager("Joji",
22)};        for(int index=0; index<vec.length; index++)
        {
            vec[index].print();
        }
```

```
        }
}
```

Question: 3 (Multiple Inteheritance)

Create multiple inheritance using interfaces. Create a class Person that derived from class Employee and Officer class. Employee class contains details( ) method and Officer class contains basic info( ) method. You can simply type display any text in details and info methods.

Question: 4 (Abstract Class)

We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

Question: 5 (Abstract class)

We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

Question: 6 (Abstract classes and interfaces)

1) Write an abstract class Shape
 – Data members: numSides
 – Constructor: initialize numSides
 – Concrete method: get method for numSides
 – Abstract methods: getArea(), getPerimeter()
2) Write a concrete subclass Rectangle
 – Data members: width, height
3) Write a concrete subclass RtTriangle
 – Data members: width, height
4) In another class, write a main method to define a Rectangle and a Triangle.

Question: 7 (Abstract classes and interfaces)