

# Java – Learn From Home

## Assignment – Chapter 3

Concept: OOPS Implementation-Part II

Objective: At the end of the assignment, participants will be able to:

- Implement abstract classes
- How to use dynamic polymorphism
- How to use Access Modifiers

### Problems:

#### Exercise 1:

1. Create a class Student that has:

Instance Variables: name, standard

Methods: abstract: getPercentage( ), static getTotalNoStudents( )

Constructors: Initialize name and class

b). Create a class ScienceStudent that inherits from Student

Instance Variables: physicsMarks, chemistryMarks, mathsMarks

Class Variable: noOfStudents

getPercentage Method: Computes and returns the percentage of marks

(Max marks for a subject :100)

c). Create a class HistoryStudent that inherits from Student

Instance Variables: historyMarks, civicsMarks (Max Marks for a subject: 100)

Class Variables: noOfStudents

getPercentage Method: Computes and returns the percentage of marks (Max marks for a subject :100)

d). Create a general class AllStudents. In this class create some history, science

students. Assign marks. Show the percentage of marks for each student.

Also find the total number of students.

// Check if the second example makes sense. If it does not, please remove

and add new example.

### Guided Solution:

**Step 1:** create an abstract class Student ,which has two properties name and standard.

**Step 2:** provide a Student class constructor to initialize name and standard.

**Step 3:** create an abstract method getPercentage() to be implemented in subclass like ScienceStudent and HistoryStudent.

**Step 4:** create a static method getTotalNumberOfStudents() which is used to return total number of students

**Step 5:** create a subclass sciencestudent which extends Student class and provide three instance variables like Physicsmarks,mathsmarks,chemistrymarks and one static variable noofstudents initialize with zero.

**Step 6:** provide a ScienceStudent constructor to initialize all the properties which are inherited from Student class and as well sciencestudent specific properties, by creating ScienceStudent class object in AllStudent class main method, inside constructor increment noOfStudents by one, such that whenever a new ScienceStudent object is created this constructor invoke and noOfStudents variable will be incremented by one, from this we can get total number of science students.

**Step 7:** implement getpercentage() to return overall percentage of the science student.

**Step 8:** implement getTotalNoOfStudents() to return noOfStudents

**Step 9:** similarly create HistoryStudents and repeat the same steps which you have done for ScienceStudent class

**Step 10:** Finally create the main class AllStudents , inside main method create two objects for sciencestudents ,while creating object pass the values for the properties using ScienceStudent constructor.

**Step 11:** And call getPercentage() on these two object and store the values in local variables.

**Step 12:** print total number of sciencestudents by calling getTotalNumber OfStudents() and their percentages by calling getPercentage().

## Exercise 2:

2. Create a Shape as an abstract class having the methods draw and area

methods, Implement the classes for different shapes like Circle, Rectangle,

Square in such way that, based on the shape the areas of the shapes to be

calculated and displayed.

Create an abstract class and its implemented classes.

Abstract Classes:

AbstractClassName: Shape

Abstract Methods: draw():void

area():double

display():void

Implemented Classes:

ClassNames: Rectangle, Circle and Square implementing the all abstract

methods with their own implementations. And constructor to initialize properties

Class Name: Rectangle

Instance Variables: length:double, breadth:double

Class Name: Circle

Instance Variables: radius:double

Class Name: Square

Instance Variables: side:double

## Guided Solution:

- Step 1:** Create an abstract class Shape and provide three abstract methods namely draw(),area(),display()
- Step 2:** Create a subclass Rectangle which extends Shape and provide two properties namely length,breadth.
- Step 3:** Use Rectangle constructor to initialize these two properties.
- Step 4:** Provide implementation for draw() to print "Rectangle Draw".
- Step 5:** Provide implementation for area() to return area.
- Step 6:** Provide implementation for display() to display area of the rectangle.
- Step 7:** Repeat the same steps for Circle and Square Class.
- Step 8:** Finally create a main class TestShape in which define a main method and create objects these subclasses like Rectangle,Circle,Square, and call methods like draw(),area(),display() on these objects.

## Exercise 3:

1. Create a class Employee and the sub classes Manager and Clerk:

Employee:

Instance Variables: name, empId, salary.

Methods: set and get methods for name, empId, getSalary, setSalary Method

Manager:

Instance Variables: type

Methods: setSalary()

Clerk:

Instance Variables: int speed, int accuracy

Methods: setSalary()

Provide proper constructors for all classes. Create a general class "MyClass". In

this class create objects of Manager, Clerk and Employee class. Set the name,

empId and salary attributes for each object, and accordingly display them.

## Guided Solution:

**Step 1:** Create an Employee class and provide instance variables like name,empid,salary, and initialize these variables using Employee class constructor and provide setters and getters for each instance variable in the Employee class.

**Step 2:** Create a subclass Manager which extends Employee class and Manager class specific property “type”, and generate getter method for this property.

**Step 3:** Using Manager class constructor initialize all the properties which are inherited from Employee class and Manager class specific properties.

**Step 4:** Create a subclass Clerk which extends Employee class and provide two specific properties “speed” and “accuracy”.And provide getters for these two properties.

**Step 5:** Using Clerk class constructor initialize all the properties which are inherited from Employee class and Clerk class specific properties.

**Step 6:** Finally in the main class MyClass define a main method , inside main method create objects for Manager class Clerk class, while creating these objects pass values to initialize the properties of Manager and Clerk class.And call methods like getName(),getEmpid(),getSalary() on these two objects.

**Step 7:** And call getType() method on Manager class object.

**Step 8:** And call getSpeed() and getAccuracy() on Clerk class Object.