

CSN-103: Fundamentals of Object Oriented Programming

Assignment 06

General Instructions:

- To submit this assignment: Create a single .zip file containing all the source code (.java) files. Rename the zip as 06_xxxxxxx where xxxxxx is your enrollment number.
- Send the zip file to oop2019.iitr@gmail.com
- Follow indentation while writing programs.
- All submissions will be checked for Plagiarism. ANY ATTEMPT TO CHEAT WILL BE SEVERELY PENALIZED.

Programming Problems

- Write a Java program which defines a base class called **Student** having three member variables viz., **ID**, **Name**, **PinCode**. Student class should have the following **overloaded** methods (not constructors):
 - Initialize(int id, String str, int pc)** which assigns **id**, **str**, and **pc** values to **ID**, **Name**, and **PinCode**, respectively.
 - Initialize(String str)** which assigns **str** to **Name**. ID and PinCode are assigned fixed constant values 19114000 and 247667, respectively.

Create a class **ResearchAssociate** derived from the **Student** class. ResearchAssociate class should declare three variables ID, PinCode, and Salary (hence, hiding ID and PinCode of superclass **Student**).

ResearchAssociate class should have the following **overloaded/overridden** methods (not constructors):

- Initialize(int id, String str, int pc, int sl)** which assigns **id**, **str**, **pc**, and **sl** values to **ID**, **Name**, **PinCode**, and **Salary** variables of the derived class, respectively. This method should also set **ID**, **Name**, and **PinCode** variables of the base class to -1, null, -1, respectively.
- Initialize(String str)** which assigns **str** to **Name**. ID, PinCode, and Salary are assigned fixed constant values 10080, 247001, and 15600, respectively. This method should also set **ID**, **Name**, and **PinCode** variables of the base class to -1, null, -1, respectively.

Define a third class in which you create two objects, one of Student class and another of ResearchAssociate class. Call different overloaded/overridden methods for each of these objects and observe the output.

- Consider the following 3 abstract classes:

```
abstract class BoxW {  
    int width;  
    abstract void printDimentions();  
}
```

```
abstract class BoxH {  
    int height;  
    abstract void printDimentions();  
}
```

```
abstract class BoxD {  
    int depth;  
    abstract void printDimentions();  
}
```

Create a class **Box** which should be able to access width, height, and depth from all the above abstract classes via inheritance. Also, you should define printDimentions() method (only once) in the **appropriate class** to print the value of the width, height, and depth in the Box class.