



COMP 20043

OOPM

LAB EXERCISE WEEK 5

```
class Person {
    String name; int age;
    void setName(String n) {
        name = n;
    }
    String getName(){
        return name;}

    void setAge(int a){
        age = a;
    }
    int getAge(){
        return age;}
}

class Main {
    public static void main(String[] args) {
        Person P = new Person();
        P.setName( "Ahmed");
        P.setAge ( 25);
        System.out.println("Person Name" + P.getName());
        System.out.println("Person Age" + P.getAge());
    }
}
```

This program features a class called Person with two instance variables, name and age, and set and get methods to set the values to the variables and return their values respectively. In the Main class, we create an object of the Person class named P, set the values of its properties using **set** methods, and call its **get** methods to print the properties to the console.

Tasks

1. Write a Java program to implement the **concept of polymorphism** using method overloading in class called **Box**. The class should have 3 instance variables called **length** and **breadth** and **height**. Also **three setValues()** methods to set initial values to the variables. The first set method has **no parameters** and should initialize the instance variables with some values. The second set method should have **three parameters** to initialize the variables using the values passed from main method. The third method should have **one parameter** which should be used to set the values for all the three class variables. The class should also have a method called as **display()** which displays the length, breadth and height of the box and also a message whether it is a normal box or a cube. Create another class that hosts the main method. Create an object of the Box class which invokes the all the three **setValues()** methods. Call the **display()** method also.
2. Create an abstract class called Animal with two variables name and age. The class should have a parameterized constructor to assign initial values to the variables using the values passed from the main method. The class should also have two get methods to return the values of the two variables. The class must have an abstract method makeSound() to display the sound according to the type of the animal. This should be implemented by the class which extends the Animal class. Create two subclasses of Animal called "Cat" and "Dog". Implement the makeSound() methods in both these classes to display the messages "Cat Meows" and "Dog Barks" respectively. Create a main class, create one object each for Cat and Dog classes and call the methods respectively
3. Create an **abstract class called Shape** that contains one data member color, two abstract methods called calculateArea() and displayinfo(), which is implemented by Circle class and Rectangle class that extends from the Shape class. Circle class should have two variables radius, and area, one parameterized constructor to give values for the variables color and radius. It should also implement calculateArea() method using formula $\text{area} = 3.14 * \text{radius} * \text{radius}$. It should implement displayinfo() to display, the color, radius and area. Rectangle class should have three variables width and height and area, one parameterized constructor to give values for the variables color, width and height. It should also implement calculateArea() method using formula $\text{area} = \text{width} * \text{height}$. It should also implement displayinfo() to display all its data. Create a main class, create one object each for Circle and Rectangle classes and call the methods respectively
4. Create an abstract class called BankAccount that provides a common structure for all types of bank accounts. It should have the data members, account id and account type, balance amount. It should have two methods to initialize the values of data members and display the value of data members. It should also include an abstract method calculateInterest() to calculate and display the interest, which is specific to each account type. Create subclasses SavingsAccount and CurrentAccount that extends BankAccount. They should provide implementations for the calculateInterest() method, tailored to the specific account types using the formula given below
 $\text{SavingsAccountInterest} = 2\% \text{ of balance amount}$, $\text{CurrentAccountInterest} = 0\% \text{ of balance amount}$.
In the Main class, create Objects of SavingsAccount and CurrentAccount, and call the get and set methods and calculateInterest() method to calculate the interest for each account type.

5. Create an abstract class called “ InputOutput class” that has two abstract methods. Void getdata() and void putdata(). Create a class Teacher that extends from InputOutputClass with attributes teacher id, teacher name, salary. Implement the methods getdata() and putdata() to assign values to the Teacher class variables and display the values respectively. Create another class Student that extends from InputOutputClass with attributes Student id, Student name, GPA. Implement the methods getdata() and putdata() to assign values to the Student class variables and display the values respectively. Create a main class , create one object each for Teacher and Employee classes and call the methods respectively