

# *Basic Programs Using Class, Objects and methods*

## **LAB - I**

**Object Oriented Programming (CSE2005)**

**Tharshith Goud**

19BCN7112

**SLOT - L6**



VIT-AP  
UNIVERSITY

**Faculty: Dr Aravapalli Rama Satish**



**Question 1.A:** Create an application named Percentages whose main() method holds two double variables. Assign values to the variables. Pass both variables to a method named computePercent() that displays the two values and the value of the first number as a percentage of the second one. For example, if the numbers are 2.0 and 5.0, the method should display a statement similar to “2.0 is 40 percent of 5.0.” Then call the method a second time, passing the values in reverse order. Save the application as Percentages.java.

### *Solution:*

```
class Percentages{
    public static void main(String[] args){
        double x=2.0;
        double y=5.0;
        computepercent(x,y);
        computepercent(x,y);
    }
    static void computepercent(double x,double y){
        System.out.println(x+" is "+(x/y)*100+" percent of "+y);
    }
}
```

### **Output:**

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac Percentages.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java Percentages
2.0 is 40.0 percent of 5.0
2.0 is 40.0 percent of 5.0
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █
```

**Question 1.B:** Modify the Percentages class to accept the values of the two doubles from a user at the keyboard. Save the file as Percentages2.java.

### *Solution:*

```
import java.util.Scanner;
class Percentages2{
    static Scanner in = new Scanner(System.in);
    public static void main(String[] args){
        computepercent();
        computepercent();
    }
    static void computepercent(){
        System.out.println("Enter the First Double Number");
        double x = in.nextDouble();
        System.out.println("Enter the Second Double Number");
        double y = in.nextDouble();
        System.out.println(x+" is "+(x/y)*100+" percent of "+y);
    }
}
```

### **Output:**

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\Percentages2.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java Percentages2
Enter the First Double Number
25
Enter the Second Double Number
35
25.0 is 71.42857142857143 percent of 35.0
Enter the First Double Number
375
Enter the Second Double Number
999
375.0 is 37.53753753753754 percent of 999.0
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █
```

**Question 2:** There are 12 inches in a foot and 3 feet in a yard. Create a class named InchConversion. Its main() method accepts a value in inches from a user at the keyboard, and in turn passes the entered value to two methods. One converts the value from inches to feet, and the other converts the same value from inches to yards. Each method displays the results with appropriate explanation. Save the application as InchConversion.java.

### *Solution:*

```
import java.util.Scanner;
class InchConversion{
    static Scanner in = new Scanner(System.in);
    public static void main(String[] args){
        System.out.println("Enter a values in terms of Inches: ");
        double n = in.nextDouble();
        IntoFeet(n);
        IntoYards(n);
    }
    static void IntoFeet(double n){
        System.out.println(n+" inches is equals to "+ n/12 + " foot");
    }
    static void IntoYards(double n){
        System.out.println(n+" inches is equals to "+ n/(12)*3 + " Yards");
    }
}
```

### **Output:**

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\InchConversion.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java InchConversion
Enter a values in terms of Inches:
60
60.0 inches is equals to 5.0 foot
60.0 inches is equals to 15.0 Yards
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █
```

### Question 3:

Assume that a gallon of paint covers about 350 square feet of wall space. Create an application with a main() method that prompts the user for the length, width, and height of a rectangular room. Pass these three values to a method that does the following:

- Calculates the wall area for a room
- Passes the calculated wall area to another method that calculates and returns the number of gallons of paint needed
- Displays the number of gallons needed
- Computes the price based on a paint price of \$32 per gallon, assuming that the painter can buy any fraction of a gallon of paint at the same price as a whole gallon
- Returns the price to the main() method

The main() method displays the final price. For example, the cost to paint a 15-by-20-foot room with 10-foot ceilings is \$64. Save the application as PaintCalculator.java.

### Solution:

```
import java.util.Scanner;
class PaintCalculator{
    static Scanner in = new Scanner(System.in);
    public static void main(String[] args){
        System.out.println("Enter the length of the Rectangular room: ");
        double l = in.nextDouble();
        System.out.println("Enter the height of the Rectangular room: ");
        double h = in.nextDouble();
        System.out.println("Enter the breadth of the Rectangular room: ");
        double b = in.nextDouble();
        double a = area(l,h,b);
        System.out.println("Area of the room's wall space is "+a+" sq.ft");
        double p = paint(a);
        print(p);
        double c = cost(p);
        System.out.println("Total cost of painting to paint a room of dimension
s "+l+"x"+h+"x"+b+" is "+c);
    }
    static double area(double l,double b, double h){
        return 2*(l*b)+2*(b*h)+2*(h*l);
    }
    static double paint(double a){
        return a/350;
    }
    static void print(double p){
        System.out.println("Total gallons of paint required is "+p);
    }
    static double cost(double p){
        return p*32;
    }
}
```

## Output:

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\PaintCalculator.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java PaintCalculator
Enter the length of the Rectangular room:
10
Enter the height of the Rectangular room:
10
Enter the breadth of the Rectangular room:
10
Area of the room's wall space is 600.0 sq.ft
Total gallons of paint required is 1.7142857142857142
Total cost of painting to paint a room of dimensions 10.0x10.0x10.0 is 54.857142857142854
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █
```

**Question 4:** Herbert's Home Repair estimates each job cost as the cost of materials plus \$35 per hour while on the job, plus \$12 per hour for travel time to the job site. Create a class that contains a main() method that prompts the user for the name of a job (for example, Smith bathroom remodel), the cost of materials, the number of hours of work required, and the number of hours travel time. Pass the numeric data to a method that computes estimate for the job and returns the computed value to the main() method where the job name and estimated price are displayed. Save the program as JobPricing.java.

### Solution:

```
import java.util.Scanner;
class JobPricing{
    static Scanner in = new Scanner(System.in);
    public static void main(String[] args){
        System.out.println("Enter the name of Job: ");
        String job = in.nextLine();
        System.out.println("Enter the cost of materials: ");
        double cost = in.nextDouble();
        System.out.println("Enter No. of hours worked: ");
        double hours = in.nextDouble();
        System.out.println("Enter Travel Time: ");
        double travel = in.nextDouble();
        double price = estimateadprice(cost, hours, travel);
        System.out.println("Total estimatead cost to finish "+job+" is $" + price);
    }
    static double estimateadprice(double c, double h, double t){
        return (c+(h*35)+(t*12));
    }
}
```

## Output:

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\JobPricing.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java JobPricing
Enter the name of Job:
Smith bathroom remodel
Enter the cost of materials:
150
Enter No. of hours worked:
4
Enter Travel Time:
2
Total estimatead cost to finish Smith bathroom remodel is $314.0
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █
```

**Question 5.A:** Create a class named Student that has fields for an ID number, number of credit hours earned, and number of points earned. (For example, many schools compute grade point averages based on a scale of 4, so a three-credit-hour class in which a student earns an A is worth 12 points.) Include methods to assign values to all fields. A Student also has a field for grade point average. Include a method to compute the grade point average field by dividing points by credit hours earned. Write methods to display the values in each Student field. Save this class as Student.java.

### *Solution:*

```
import java.util.Scanner;
class Student1{
    String ID;
    double c;
    double p;
    double gpa;
    void setID(String ID){
        this.ID = ID;
    }
    void setc(double c){
        this.c = c;
    }
    void setp(double p){
        this.p = p;
    }
    void calculategpa(){
        this.gpa = p/c;
    }
}
```

```

void displayID(){
    System.out.println("Your ID is "+ID);
}
void displaycred(){
    System.out.println("Your total credits is "+c);
}
void displayp(){
    System.out.println("No of points earned is "+p);
}
void displaygpa(){
    System.out.println("Your GPA is "+gpa);
}
}
class Student{
    public static void main(String[] args){
        Student1 ob = new Student1();
        ob.setID("19BCN7112");
        ob.setc(20);
        ob.setp(76);
        ob.calculategpa();
        ob.displayID();
        ob.displaycred();
        ob.displayp();
        ob.displaygpa();
    }
}

```

## Output:

```

PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\Student.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java Student
Your ID is 19BCN7112
Your total credits is 20.0
No of points earned is 76.0
Your GPA is 3.8
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> 

```

**Question 5.B:** Write a class named ShowStudent that instantiates a Student object from the class you created and assign values to its fields. Compute the Student grade point average, and then display all the values associated with the Student. Save the application as ShowStudent.java.



***Solution:*** (The code for 5a and 5b is almost same)

```
class Student{
    String ID;
    double c;
    double p;
    double gpa;
    void setID(String ID){
        this.ID = ID;
    }
    void setc(double c){
        this.c = c;
    }
    void setp(double p){
        this.p = p;
    }
    void calculategpa(){
        this.gpa = p/c;
    }
    void displayID(){
        System.out.println("Your ID is "+ID);
    }
    void displaycred(){
        System.out.println("Your total credits is "+c);
    }
    void displayp(){
        System.out.println("No of points earned is "+p);
    }
    void displaygpa(){
        System.out.println("Your GPA is "+gpa);
    }
}

class ShowStudent{
    public static void main(String[] args){
        Student ob = new Student();
        ob.setID("19BCN7112");
        ob.setc(20);
        ob.setp(76);
        ob.calculategpa();
        ob.displayID();
        ob.displaycred();
        ob.displayp();
        ob.displaygpa();
    }
}
```

## Output:

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\Student.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java Student
Your ID is 19BCN7112
Your total credits is 20.0
No of points earned is 76.0
Your GPA is 3.8
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> []
```

**Question 5.C:** Create a constructor for the Student class you created. The constructor should initialize each Student's ID number to 9999, his or her points earned to 12, and credit hours to 3 (resulting in a grade point average of 4.0). Write a program that demonstrates that the constructor works by instantiating an object and displaying the initial values. Save the application as ShowStudent2.java.

## *Solution:*


```
class Student{
    String ID;
    double c;
    double p;
    double gpa;
    Student(){
        this.ID = "9999";
        this.c=3;
        this.p=12;
    }
    void setID(String ID){
        this.ID = ID;
    }
    void setc(double c){
        this.c = c;
    }
    void setp(double p){
        this.p = p;
    }
    void calculategpa(){
        this.gpa = p/c;
    }
    void displayID(){
        System.out.println("Your ID is "+ID);
    }
}
```

```

    void displaycred(){
        System.out.println("Your total credits is "+c);
    }
    void displayp(){
        System.out.println("No of points earned is "+p);
    }
    void displaygpa(){
        System.out.println("Your GPA is "+gpa);
    }
}
class ShowStudent2{
    public static void main(String[] args){
        Student ob = new Student();
        ob.calculategpa();
        ob.displayID();
        ob.displaycred();
        ob.displayp();
        ob.displaygpa();
    }
}

```

## Output:



```

PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\ShowStudent2.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java ShowStudent2
Your ID is 9999
Your total credits is 3.0
No of points earned is 12.0
Your GPA is 4.0
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █

```

## Question 6.A:

Create a class named Lease with fields that hold an apartment tenant's name, apartment number, monthly rent amount, and term of the lease in months. Include a constructor that initializes the name to "XXX", the apartment number to 0, the rent to 1000, and the term to 12. Also include methods to get and set each of the fields. Include a nonstatic method named addPetFee() that adds \$10 to the monthly rent value and calls a static method named explainPetPolicy() that explains the pet fee. Save the class as Lease.java.

### *Solution:*

```
class Lease{
    String tennant;
    int aptNo;
    double rent;
    int term;
    Lease(){
        tennant = "XXX";
        aptNo = 0;
        rent = 1000;
        term = 12;
    }
    public String getTennant() {
        return tennant;
    }
    public void setTennant(String tennant) {
        this.tennant = tennant;
    }
    public int getAptNo() {
        return aptNo;
    }
    public void setAptNo(int aptNo) {
        this.aptNo = aptNo;
    }
    public double getRent() {
        return rent;
    }
    public void setRent(double rent) {
        this.rent = rent;
    }
    public int getTerm() {
        return term;
    }
    public void setTerm(int term) {
```

```

        this.term = term;
    }
    public void addPetFee(){
        this.rent = rent+10;
    }
    public static void explainPetPolicy(){
        System.out.println("You are Having pet So you need to pay a $10 premium");
    }
}

```

**Output:**

*No Output*

**Question 6.B:** Create a class named TestLease whose main() method declares four Lease objects. Call a getData() method three times. Within the method, prompt a user for values for each field for a Lease, and return a Lease object to the main() method where it is assigned to one of main()'s Lease objects. Do not prompt the user for values for the fourth Lease object, but let it continue to hold the default values. Then, in main(), pass one of the Lease objects to a showValues() method that displays the data. Then call the addPetFee() method using the passed Lease object and confirm that the fee explanation statement is displayed. Next, call the showValues() method for the Lease object again and confirm that the pet fee has been added to the rent. Finally, call the showValues() method with each of the other three objects; confirm that two hold the values you supplied as input and one holds the constructor default values. Save the application as TestLease.java.

## Solution:

```
import java.util.Scanner;

class TestLease{
    static Scanner in = new Scanner(System.in);
    public static void main(String[] ar){
        Lease ob1 = new Lease();
        Lease ob2 = new Lease();
        Lease ob3 = new Lease();
        Lease ob4 = new Lease();
        getData(ob1);
        getData(ob2);
        getData(ob3);
        showValues(ob1);
        ob1.addPetFee();
        ob1.explainPetPolicy();
        showValues(ob1);
        showValues(ob2);
        showValues(ob3);
        showValues(ob4);

    }
    public static void getData(Lease ob){
        System.out.print("Input the name of tenant: ");
        String name = in.nextLine();
        System.out.print("Input the Apartment number: ");
        int apt = in.nextInt();
        System.out.print("Input the Monthly Rent: ");
        double rent = in.nextDouble();
        System.out.print("Input Total Term in Months: ");
        int term = in.nextInt();
        ob.setTennant(name);
        ob.setAptNo(apt);
        ob.setRent(rent);
        ob.setTerm(term);
        in.nextLine();
    }
    public static void showValues(Lease ob){
        System.out.println("The Name of Tennant is "+ob.tennant+" He lives in Apartment number "+ob.aptno+. His Monthly rent is "+ ob.rent+" his lease term is "+ob.term);
    }
}
```

## Output:

```
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> javac .\TestLease.java
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> java TestLease
Input the name of tenant: Obul Reddy
Input the Apartment number: 405
Input the Monthly Rent: 500
Input Total Term in Months: 12
Input the name of tenant: Lionel Messi
Input the Apartment number: 112
Input the Monthly Rent: 850
Input Total Term in Months: 36
Input the name of tenant: Lewis Hamilton
Input the Apartment number: 711
Input the Monthly Rent: 1050
Input Total Term in Months: 42
The Name of Tennant is Obul Reddy He lives in Apartment number 405. His Monthly rent is 500.0 his lease term is 12
You are Havihg pet So you need to pay a $10 premium
The Name of Tennant is Obul Reddy He lives in Apartment number 405. His Monthly rent is 510.0 his lease term is 12
The Name of Tennant is Lionel Messi He lives in Apartment number 112. His Monthly rent is 850.0 his lease term is 36
The Name of Tennant is Lewis Hamilton He lives in Apartment number 711. His Monthly rent is 1050.0 his lease term is 42
The Name of Tennant is XXX He lives in Apartment number 0. His Monthly rent is 1000.0 his lease term is 12
PS C:\Users\Dracarys\Desktop\LABS\OOP\19BCN7112> █
```

## GITHUB LINK:

<https://github.com/tharshith44/OOPLab>



VIT-AP  
UNIVERSITY

THE END