**Department of Computing**

**CS 212: Object Oriented Programming**

**Class: BESE-11AB**

**Lab 07: Fundamentals of OOP**

**Date: April 14th, 2021**

**Instructor:**

Ms. Hania Aslam

# Learning Objectives

The learning objectives of this lab is to understand and practice the fundamentals of object-oriented programming, such as classes and objects, constructors & method invocation.

# Activity #1.

What is wrong in the following code snippets? Please rectify each, and provide updated results along with a brief explanation in a separate file.

|  |
| --- |
| public class Test1  {  public static void main(String[] args)  {  Test1 t1 = new Test1(5);  }  }  **Explanation:**  The above code gives error because we have passed a parameter to the constructor during object creation where as there is no parameterized constructor defined in the class and hence the error. This error can be rectified by either removing the int value ‘5’ from the constructor or define a parameterized constructor for the code to run fine. |
| public class Test2  {  public static void main(String[] args)  {  Test2 t2 = new Test2(); t2.x();  }  }  **Explanation:**  The above code gives error because we have not defined the method named ‘x’ in the class. This error can be rectified by defining a method ‘x’ in the class. |
| public class Test3  {  Public void method1()  {  Circle c;  System.out.println(“Radius is ” + c.getRadius()); c = new Circle();  }  }  **Explanation:**  The above code gives error because at the time we call the method of the Circle class ‘getRadius()’ the object of the class ‘c’ is not created and is created after invoking the method. This error can be rectified by creating the object of the Circle class ‘c’ before calling the method ‘getRadius()’ i.e., Circle c = new Circle(); |
| public class Test4  {  public static void main(String[] args)  {  C c = new C(5.0); System.out.println(c.value);  }  }  class C  {  int value = 2;  }  **Explanation:**  The above code gives error because we have passed a parameter to the constructor during object creation where as there is no parameterized constructor defined in the class ‘C’ and hence the error. The error can be rectified by removing the double value ‘5.0’ from the constructor and hence the output of the above code will be ‘2’. |

# Activity #2.

What is the difference between passing a parameter of a primitive type and passing a parameter of a reference type? Show the output of the following program after analyzing it(*you may provide your answer and the output in commented form at the top of the program’s source file*):

|  |
| --- |
| // save in a file named Count.java public class Count  {  public int count;  public Count(int c)  {  count = c;  }  public Count()  {  count = 1;  }  } |

|  |
| --- |
| // save in a file named Test.java  public class Test  {  public static void main(String[] args)  {  Count myCount = new Count(); int times = 0;  for (int i = 0; i < 100; i++) increment(myCount, times);  System.out.println("count is " + myCount.count); System.out.println("times is " + times);  }  public static void increment(Count c, int times)  {  c.count++; times++;  }  } |

**Output:**

count is 101

times is 0

**Explanation:**

When we pass a parameter of primitive type, we actually pass a copy of value of that variable. But when we pass a parameter of reference type, we pass the reference value of that object or variable i.e., the memory address of that object or variable. As it is not a copied value, we can change value of the object or variable but it cannot be done in the case of primitive types.

# Task #1:

Design a class named Rectangle to represent a rectangle. The class contains:

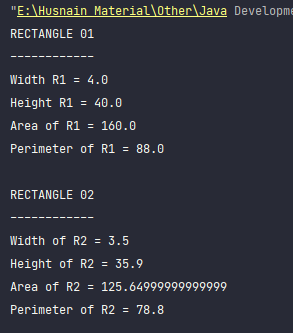
* Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.
* A no-arg constructor that creates a default rectangle.
* A constructor that creates a rectangle with the specified width and height.
* A method named getArea() that computes and returns the area of this rectangle.
* A method named getPerimeter() that returns the perimeter.

Implement the class. Write a test program that creates two Rectangle objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order.

**Code:**

package com.company;  
  
*//Creating our own class Rectangle*class Rectangle{  
 *//Declaring fields* double width = 1;  
 double height = 1;  
  
 *//Constructor for creating default rectangle* public Rectangle(){  
  
 }  
 *//Constructor for initializing the variables* public Rectangle(double *w*, double *h*){  
 width = *w*;  
 height = *h*;  
 }  
 *//Method for calculating the area of rectangle* public double getArea(double *width*, double *height*){  
 return *width* \* *height*;  
 }  
 *//Method for calculating the perimeter of rectangle* public double getPerimeter(double *width*, double *height*){  
 return 2 \* (*width* + *height*);  
 }  
}  
  
public class Task01 {  
  
 *//Main method* public static void main(String[] *args*) {  
  
 *//Data of rectangle 1* System.*out*.println("RECTANGLE 01");  
 System.*out*.println("------------");  
 *//Creating an object of Rectangle class* Rectangle r1 = new Rectangle(4,40);  
 *//Displaying required outputs* System.*out*.println("Width R1 = " + r1.width);  
 System.*out*.println("Height R1 = " + r1.height);  
 System.*out*.println("Area of R1 = " + r1.getArea(4,40));  
 System.*out*.println("Perimeter of R1 = " + r1.getPerimeter(4,40));  
  
 System.*out*.println();  
  
 *//Data of rectangle 2* System.*out*.println("RECTANGLE 02");  
 System.*out*.println("------------");  
 *//Creating another object of Rectangle class* Rectangle r2 = new Rectangle(3.5,35.9);  
 *//Printing required outputs* System.*out*.println("Width of R2 = " + r2.width);  
 System.*out*.println("Height of R2 = " + r2.height);  
 System.*out*.println("Area of R2 = " + r2.getArea(3.5,35.9));  
 System.*out*.println("Perimeter of R2 = " + r2.getPerimeter(3.5,35.9));  
  
 }  
}

**Output Screenshot:**

****

# Task #2:

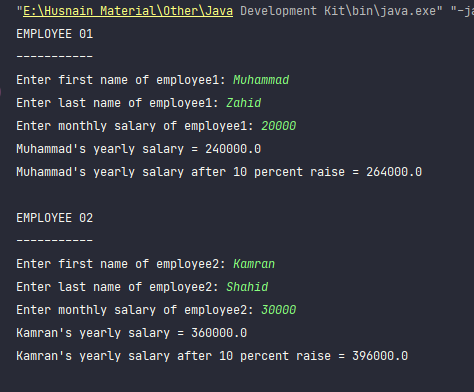
Create a class called Employee that includes three instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Provide a constructor that initializes the three variables. If, monthly salary provided by the user is not positive, it should be set to 0 instead.

Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

**Code:**

package com.company;  
  
import java.util.Scanner;  
  
*//Creating our own class Employee*class Employee{  
 *//Declaring fields* String first\_name;  
 String last\_name;  
 double monthly\_salary;  
  
 *//Constructor for initializing the variables* public Employee(String *f\_n*, String *l\_n*, double *m\_s*){  
 first\_name = *f\_n*;  
 last\_name = *l\_n*;  
 monthly\_salary = *m\_s*;  
  
 if (monthly\_salary < 0){  
 monthly\_salary = 0;  
 }  
 }  
 *//Method to calculate yearly salary* public double yearlySalary(){  
 return monthly\_salary \* 12;  
 }  
 *//Method to calculate 10% raise in salary* public double bonus(){  
 return monthly\_salary \* 0.1;  
 }  
}  
  
public class Task02 {  
  
 *//Main method* public static void main(String[] *args*) {  
  
 *//Creating an object of Scanner class to take inputs* Scanner sc = new Scanner(System.*in*);  
  
 *//Taking the inputs for employee 1* System.*out*.println("EMPLOYEE 01");  
 System.*out*.println("-----------");  
 System.*out*.print("Enter first name of employee1: ");  
 String first\_name\_1 = sc.next();  
 System.*out*.print("Enter last name of employee1: ");  
 String last\_name\_1 = sc.next();  
 System.*out*.print("Enter monthly salary of employee1: ");  
 double monthly\_salary\_1 = sc.nextDouble();  
  
 *//Creating an object of Employee class* Employee person1 = new Employee(first\_name\_1, last\_name\_1, monthly\_salary\_1);  
 *//Printing required outputs* System.*out*.printf("%s's yearly salary = %.1f\n", person1.first\_name, person1.yearlySalary());  
 System.*out*.printf("%s's yearly salary after 10 percent raise = %.1f\n", person1.first\_name, (person1.yearlySalary() + (12 \* person1.bonus())));  
  
 System.*out*.println();  
  
 *//Taking inputs for employee 2* System.*out*.println("EMPLOYEE 02");  
 System.*out*.println("-----------");  
 System.*out*.print("Enter first name of employee2: ");  
 String first\_name\_2 = sc.next();  
 System.*out*.print("Enter last name of employee2: ");  
 String last\_name\_2 = sc.next();  
 System.*out*.print("Enter monthly salary of employee2: ");  
 double monthly\_salary\_2 = sc.nextDouble();  
  
 *//Making another object of Employee class* Employee person2 = new Employee(first\_name\_2, last\_name\_2, monthly\_salary\_2);  
 *//Printing required outputs* System.*out*.printf("%s's yearly salary = %.1f\n", person2.first\_name, person2.yearlySalary());  
 System.*out*.printf("%s's yearly salary after 10 percent raise = %.1f\n", person2.first\_name, (person2.yearlySalary() + (12 \* person2.bonus())));  
 }  
}

**Output Screenshot:**

****

# Task #3:

Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a constructor that initializes the three variables and assumes that the values provided are correct. Provide a method displayDate that displays the month, day and year separated by forward slashes ( / ). Write a test application named DateTest that demonstrates class Date’s capabilities.

**Code:**

package com.company;  
  
import java.util.Scanner;  
  
*//Creating our own class Date*class Date{  
 *//Declaring fields* int month;  
 int day;  
 int year;  
  
 *//Constructor for initializing the variables* public Date(int *m*, int *d*, int *y*){  
 month = *m*;  
 day = *d*;  
 year = *y*;  
 }  
 *//Method to display date to the console* public void displayDate(){  
 System.*out*.printf("%d/%d/%d", month, day, year);  
 }  
}  
  
public class Task03 {  
  
 *//Main method* public static void main(String[] *args*) {  
  
 *//Creating an object of Scanner class to take inputs* Scanner sc = new Scanner(System.*in*);  
  
 *//Taking inputs* System.*out*.print("Enter the month: ");  
 int month = sc.nextInt();  
 System.*out*.print("Enter the day: ");  
 int day = sc.nextInt();  
 System.*out*.print("Enter the year: ");  
 int year = sc.nextInt();  
  
 *//Creating an object of Date class* Date today = new Date(month, day, year);  
 *//Printing the date* today.displayDate();  
  
 }  
}

# Output Screenshot:

# 

# Hand in

Hand in the source code from this lab at the appropriate location on the LMS system.

# To Receive Credit

1. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.
2. The lab time is not intended as free time for working on your programming/other assignments.