LAB 03 : TASK

Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'

class Employee

{

String name;

int year\_of\_joining;

double salary;

String address;

void setData(String Name, int Year\_of\_joining, String Address)

{

name=Name;

year\_of\_joining=Year\_of\_joining;

address=Address;

}

void Display()

{

System.out.println(name+"\t"+year\_of\_joining+"\t\t"+address);

}

}

class Exercise1

{

public static void main(String args[])

{

Employee e1=new Employee();

Employee e2=new Employee();

Employee e3=new Employee();

e1.setData("Robert", 1994 , "64C- WallsStreat");

e2.setData("Sam" , 2000 , "68D- WallsStreat");

e3.setData("John" , 1999 , "26B- WallsStreat.");

System.out.println("Name\tYear of Joining\tAddress");

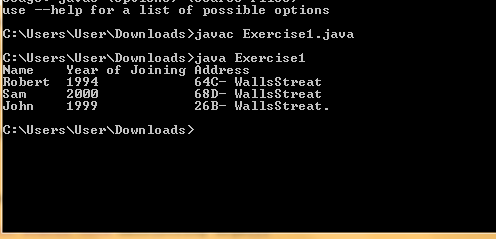
e1.Display();

e2.Display();

e3.Display();

}

}



Write a program by creating an 'EmployeeInfo' class having the following methods and print the final salary. 1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as parameter 2 - 'AddWork()' which adds $5 to salary of employee if the number of hours of work per day is more than 6 hours.

import java.util.Scanner;

class infor

{

double salary;

int day;

void getinfo(double salary,int day)

{

this.salary=salary;

this.day=day;

}

void Addwork()

{

if(day>6)

System.out.println(salary+5);

else

System.out.println(salary);

}

} class EmployeeInfo {

public static void main(String args[])

{

double salary;

int day;

infor fo = new infor();

Scanner sc=new Scanner(System.in);

System.out.println("Enter salary");

salary=sc.nextDouble();

System.out.println("Enter hour of work perday");

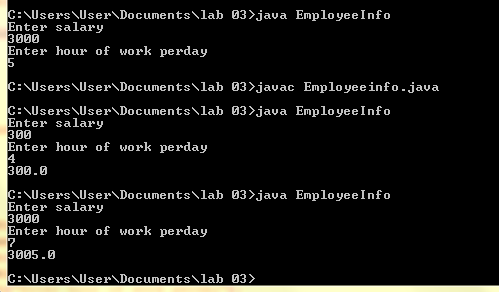
day=sc.nextInt();

fo.getinfo(salary,day);

fo.Addwork();

}

}



Create a class Vehicle. The class should have two fields-no\_of\_seats and no\_of\_wheels. Create two objects-Motorcycle and Car for this class. Your output should show the descriptions for Car and Motorcycle.

class Vehicle

{

int no\_of\_seats;

int no\_of\_wheels;

Vehicle()

{

no\_of\_seats=2;

no\_of\_wheels=2;

}

Vehicle(int a,int b)

{

no\_of\_seats=a;

no\_of\_wheels=b;

}

}

class Exercise3

{

public static void main(String args[])

{ Vehicle motorcycle=new Vehicle();

Vehicle car=new Vehicle(5,4);

System.out.println("Car contain "+car.no\_of\_wheels+" wheels");

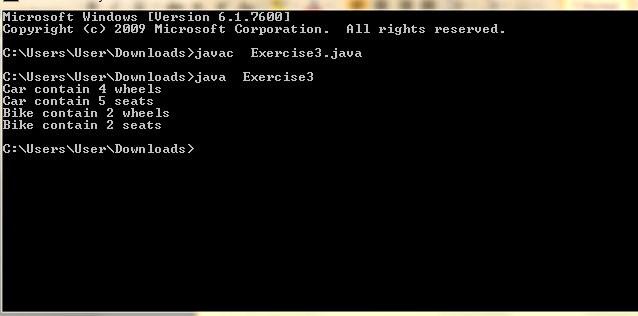
System.out.println("Car contain "+car.no\_of\_seats+" seats");

System.out.println("Bike contain "+motorcycle.no\_of\_wheels+" wheels");

System.out.println("Bike contain "+motorcycle.no\_of\_seats+" seats");

}

}



Write a program that inputs temperature in Celsius and converts it into Fahrenheit.

import java.util.Scanner;

public class Temprature {

public static void main(String[] Strings) {

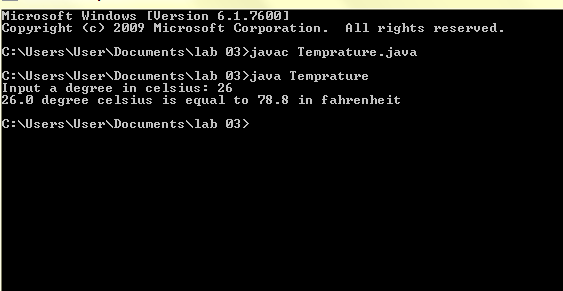
Scanner input = new Scanner(System.in);

System.out.print("Input a degree in celsius: ");

double celsius = input.nextDouble();

double fahrenheit =(( (celsius \*9/5 )+32.0 ));

System.out.println(celsius + " degree celsius is equal to " + fahrenheit + " in fahrenheit");

}}}

Write a program that inputs the year a person is born in and returns the age of the person.

import java.util.Scanner;

public class Year {

public static void main(String[] Strings) {

Scanner input = new Scanner(System.in);

System.out.println("Enter the year");

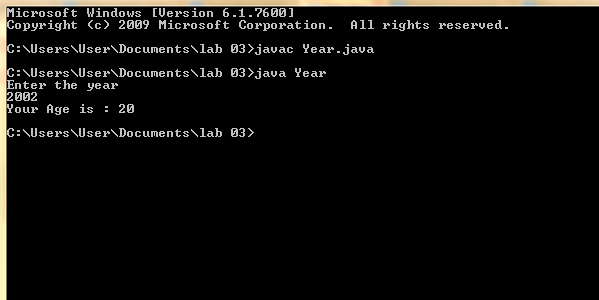
int year=input.nextInt();

int result=2022-year;

System.out.println("Your Age is : " + result);

}

}



**Post lab questions**

1.Can constructors be private?

Yes, we can declare a constructor as private. If we declare a constructor as private we are not able to create an object of a class. We can use this private constructor in the Singleton Design Pattern.

2.Can a non-static method access a static variable or call a static method?

Yes, non-static methods can access any static method and static variable also, without using the object of the class

3.State differences between java constructor and java methods?

|  |  |
| --- | --- |
| **Constructors** | **Methods** |
| A Constructor is a block of code that initializes a newly created object. | | A Method is a collection of statements which returns a value upon its execution. |
| A Constructor can be used to initialize an object. | | A Method consists of Java code to be executed |