Date : 20/11/2022
Roll No. and Name : 22BCE538 Shah Kaivan
Course Code and Name : 2CS302 Object Oriented Programming

Practical No.: 4(a)

Aim: Write a Java program by creating an 'Employee' class having the following methods and print the final salary.

• 'getInfo()' which takes the salary, number of hours of work per day of employee as parameter

• 'AddSal()' which adds $10 to salary of the employee if it is less than $500.

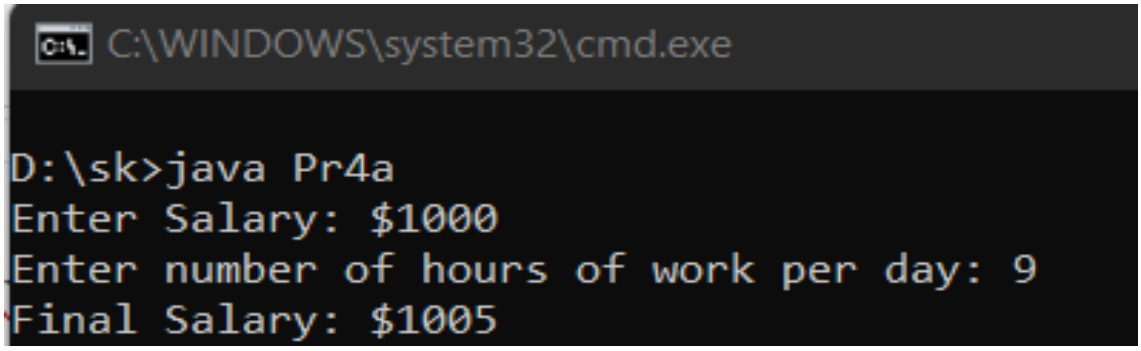
• 'AddWork()' which adds $5 to salary of employee if the number of hours of work per day is more than 6 hours.

Methodology followed:
Input:

```
import java.util.*;

class Employee

{

   int salary, hours;

   void getInfo()

        {

           Scanner sc = new Scanner(System.in);

           System.out.print("Enter Salary: $");

           salary = sc.nextInt();

           System.out.print("Enter number of hours of work per day: ");

           hours = sc.nextInt();

        }

        void AddSal()

        {

           if(salary<500)
```

```
            salary+=10;
        }
        void AddWork()
        {
           if(hours>6)
            salary+=5;
        }
        void Display()
        {
           System.out.print("Final Salary: $" + salary);
        }
}
public class Pr4a
{
        public static void main(String[] args) {



           Employee e = new Employee();
           e.getInfo();
           e.AddSal();
           e.AddWork();
           e.Display();
        }
}
```

Output:



Conclusion :

By this Practical I learnt how to access different class's data using object.

Practical No.: 4(b)

Aim: Write above program using concepts of constructor and parameterized constructor.

Methodology followed:

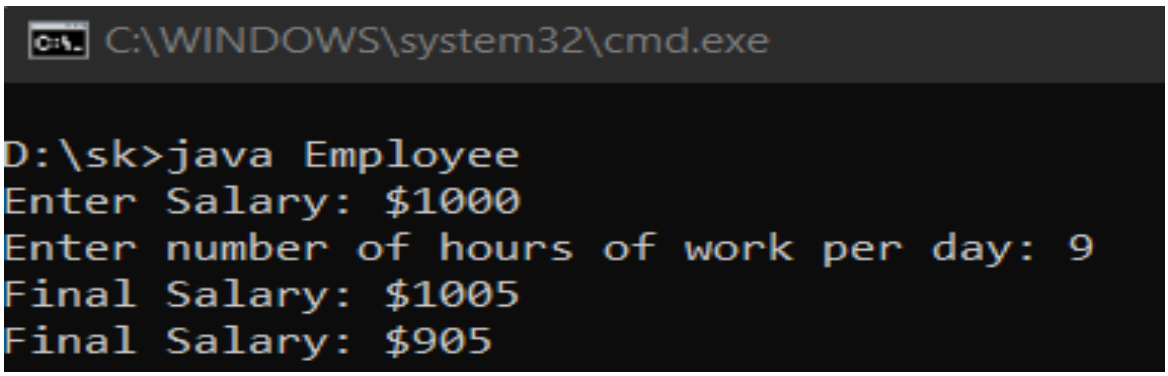
Input:

```
import java.util.*;

class Employee

{

        int salary, hours;

        Employee()

        {

                getInfo();

                AddSal();

                AddWork();

                Display();

        }

        Employee(int s, int h)

        {

                salary = s;

                hours = h;

                AddSal();

                AddWork();

                Display();

        }
```

```
void getInfo()
{
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Salary: $");
        salary = sc.nextInt();
        System.out.print("Enter number of hours of work per day: ");
        hours = sc.nextInt();
 }
void AddSal()
{
        if(salary<500)
                salary+=10;
}
void AddWork()
{
        if(hours>6)
                salary+=5;
}
void Display()
{
        System.out.println("Final Salary: $" + salary);
}
```

```
    public static void main(String[] args)
    {
        Employee e1 = new Employee();
        Employee e2 = new Employee(900,9);
    }
}
```

Output:



Conclusion:

By this practical I learnt how to use a default and parameterized constructor in java.

Practical No.: 4(c)

Aim: Create a class called complex for performing arithmetic operations with complex numbers. Use floating point variables to represent the private data of the class. Provide a default constructor that initializes the object with some default values. Provide public member methods for each of the following:

• Addition of two complex numbers: It returns the result obtained by adding the respective real parts and the imaginary parts of the two complex numbers. The method must return complex class object.

• Subtraction of two complex numbers: It returns the result obtained by subtracting the respective real parts and the imaginary parts of the two complex numbers. The method must return complex class object.

• display( ) – It displays the complex number in a + bi format. The output should be displayed as follows:- Sum of a1+b1i & a2+b2i is : a3+b3i.

Methodology followed:

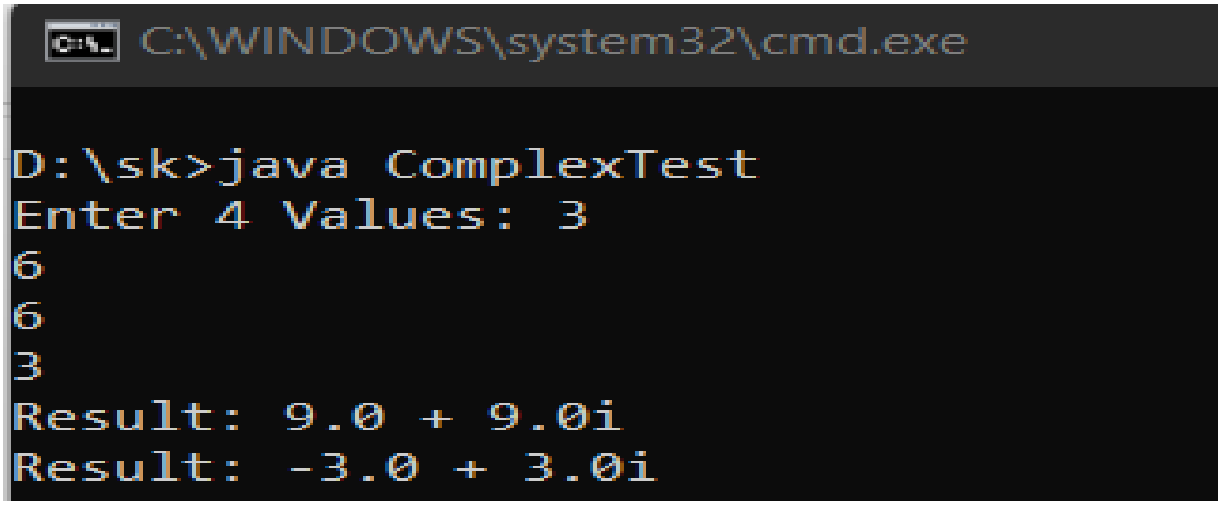
Input:

```
import java.util.*;

class Complex

{

        private float real;

        private float img;

        Complex()

   {

      real = 1;

      img = 1;

   }

        Complex(int r,int i)

        {

                real = r;
```

```
        img = i;
    }
    public Complex Addition(Complex c1, Complex c2)
    {
        Complex c = new Complex();
        c.real = c1.real+c2.real;
        c.img = c1.img + c2.img;
        return c;
    }
    public Complex Subtraction(Complex c1,Complex c2)
    {
        Complex c = new Complex();
        c.real = c1.real-c2.real;
        c.img = c1.img-c2.img;
        return c;
    }
    public void Display()
    {
        if(img>0)
        {
            System.out.println("Result: "+real+" + "+img+"i");
        }
        else
        {
```

```
                System.out.println("Result: "+real+" "+img+"i");
            }
        }
}
class ComplexTest
{
        public static void main(String args[])
        {
            Scanner scan = new Scanner(System.in);
            int n1,n2,n3,n4;
            System.out.print("Enter 4 Values: ");
            n1 = scan.nextInt();
            n2 = scan.nextInt();
            n3 = scan.nextInt();
            n4 = scan.nextInt();
            Complex c1 = new Complex(n1,n2);
            Complex c2 = new Complex(n3,n4);
            Complex c3 = c1.Addition(c1,c2);
            c3.Display();
            Complex c4 = c1.Subtraction(c1,c2);
            c4.Display();

        }
}
```

Output:



Conclusion:

By this Practical I learnt what is access specifer & how to use & access it.