

Data Structures

(UNC402)

Submitted by:
Name:- Aarav Gupta
Roll No:-
1024060206
Subgroup:- 2F31 (ECE)



Submitted to:-
Dr. Poonam Verma Bhardwaj Ma'am

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,
(DEEMED TO BE UNIVERSITY),**
**PATIALA, PUNJAB
INDIA**

Lab Assignment 2

UNC402_Data Structures

Programs based on classes and objects in C++

1. Temperature Conversion

Write a C++ program using a class that will ask for a temperature in Celsius and display it in degree Fahrenheit. The formula for conversion is $[F = 9C/5 + 32]$. You should create a class name 'Temperature' with the following members:

- Data Member: float Celsius
- Member Functions:
 - void getCelsius() - to input the temperature in Celsius.
 - float toFahrenheit() - to convert the temperature to Fahrenheit.
 - void displayFahrenheit() - to display the temperature in Fahrenheit

2. Student Details

Write a C++ program using a class to store and manage the details of students. The class should have the following data members and member functions:

Data Members (properties):

- Name
- Roll No
- Degree
- Hostel
- CGPA

Member Functions (behavior):

- ***void addDetails()*** - to input student details.
- ***void updateDetails()*** - to update all student details.
- ***void updateCGPA(float newCGPA)*** - to update the student's CGPA.
- ***void updateHostel(string newHostel)*** - to update the hostel information.
- ***void displayDetails()*** - to display student details.

3. Complex Number Operations

Construct a class named 'Complex' with private data members (real and imaginary) and the following member functions:

- ***void set(float r, float i)*** - to initialize object values.
- ***void display()*** - to display the complex number.
- ***Complex sum(Complex c)*** - to add two complex numbers and return the result as a Complex object.

Ensure that the properties are private (real and imaginary), and the member functions are public

4. Object as Argument and Return

Write a C++ program to demonstrate passing an object as an argument and returning the object from a function. Use either pass-by-value or pass-by-reference. Create a class 'Example' with the following members:

- **Data Member:** *int data*

- **Member Functions:**

- ***void setData(int value)*** - to set the value of data.
- ***void displayData()*** - to display the value of data.
- ***Example returnObjectByValue()*** - to return an object with modified data (using pass-by-value).
- ***void modifyObjectByReference(Example &obj)*** - to modify an object using pass-by-reference.

```
import java.util.Scanner;

public class Question1 {
    // Temperature Conversion
    public static void main(String[] args) {
        Temperature temp = new Temperature();

        temp.getcelsius();
        temp.toFarhenheit();
        temp.displayFahrenheit();
    }

    public static class Temperature {
        float celsius;
        float fahrenheit;

        void getcelsius() {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the temperature in celsius: ");
            celsius = scanner.nextFloat();
        }

        float toFarhenheit() {
            fahrenheit = (9.0f/5.0f * celsius) + 32.0f;
            return fahrenheit;
        }

        void displayFahrenheit() {
            System.out.println("Temperature in Farenheit: " + fahrenheit);
        }
    }
}
```

```
import java.util.Scanner;

public class Question2 {
    public static void main(String[] args) {
        // 1. Create the object
        Student s1 = new Student();

        // 2. Call methods
        System.out.println("Add Student Details");
        s1.addDetails();

        System.out.println("\n Displaying Details");
        s1.displayDetails();

        System.out.println("\n Updating CGPA");
        s1.updateCGPA(9.5f);
        s1.displayDetails();

        System.out.println("\n Updating Hostel");
        s1.updateHostel("Neeram");
        s1.displayDetails();
    }

    public static class Student {
        String name;
        int rollNumber;
        String degree;
        String hostel;
        float cgpa;

        Scanner scanner = new Scanner(System.in);

        // to input student details.
        void addDetails() {
            System.out.println("Enter student name: ");
            name = scanner.nextLine();

            System.out.println("Enter student roll number: ");
            rollNumber = scanner.nextInt();

            System.out.println("Enter student Degree: ");
            degree = scanner.nextLine();
        }
    }
}
```

```
System.out.println("Enter student Hostel: ");
hostel = scanner.nextLine();

System.out.println("Enter student CGPA: ");
cgpa = scanner.nextFloat();
}

// to update all student details.
void updateDetails() {
    // rerun the add details function and override the values
    System.out.println("Updating all details...");
    addDetails();
}

// to update the student's CGPA.
void updateCGPA(float newCGPA) {
    cgpa = newCGPA;
    System.out.println("CGPA updated successfully.");
}

// to update the hostel information.
void updateHostel(String newHostel) {
    hostel = newHostel;
    System.out.println("Hostel updated successfully.");
}

// to display student details.
void displayDetails() {
    System.out.println("Name: " + name);
    System.out.println("Roll No: " + rollNumber);
    System.out.println("Degree: " + degree);
    System.out.println("Hostel: " + hostel);
    System.out.println("CGPA: " + cgpa);
}
}
```

```
import java.util.Scanner;

public class Question3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Complex c1 = new Complex();
        System.out.println("First Complex Number");
        System.out.print("Enter real value: ");
        float r1 = scanner.nextFloat();
        System.out.print("Enter imaginary value: ");
        float i1 = scanner.nextFloat();
        c1.set(r1, i1);

        Complex c2 = new Complex();
        System.out.println("\nEnter Second Complex Number");
        System.out.print("Enter real value: ");
        float r2 = scanner.nextFloat();
        System.out.print("Enter imaginary value: ");
        float i2 = scanner.nextFloat();
        c2.set(r2, i2);

        Complex c3 = c1.sum(c2);

        System.out.println("\n Result");
        c3.display();
    }

    public static class Complex {
        private float real;
        private float imaginary;

        // to initialize object values
        public void set(float r, float i) {
            real = r;
            imaginary = i;
        }

        // to display the complex number
        public void display() {
            public void display() {
                System.out.println(real + " + " + imaginary + "i");
            }
        }
    }
}
```

```
// to add two complex numbers and return the result as a Complex object.
public Complex sum(Complex c) {
    Complex temp = new Complex();

    temp.real = this.real + c.real;
    temp.imaginary = this.imaginary + c.imaginary;

    return temp;
}
}
```

```

public class Question4 {
    // Object as Argument and Return
    public static void main(String[] args) {
        Example orig = new Example();
        orig.setData(10);
        orig.displayData();

        Example newObj = orig.returnObjectByValue();
        System.out.print("Original: ");
        orig.displayData();
        System.out.print("New Object: ");
        newObj.displayData();

        orig.modifyObjectByReference(orig);
        orig.displayData();
    }
}

class Example {
    int data;

    // to set the value of data.
    public void setData(int data) {
        this.data = data;
    }

    // to display the value of data.
    public void displayData() {
        System.out.println("Data is currently: " + this.data);
    }

    // to return an object with modified data (using pass-by-value).
    Example returnObjectByValue() {
        Example temp = new Example();
        temp.data = 100;
        return temp;
    }

    // to modify an object using pass-by-reference.
    void modifyObjectByReference(Example obj) {
        obj.data = 500;
    }
}

```