About Instructor

Ali Sher Kashif
Lecturer
Department of Computer Science
COMSATS University Islamabad,
Sahiwal Campus, Pakistan
Email: ali.sher@cuisahiwal.edu.pk

Agenda

- Understanding Classes
- Access Specifiers
- Creating and Using Objects
- Executing Member Functions
- Program Example
- Marks Class Example

Classes

- A class is a description of a number of similar objects
- A collection of objects with same properties and functions is known as class.
- A class is used to define the characteristics of the objects.
- It is used as a model for creating different objects of same type
- Each object of a class is known as an **instance** of its class
- **Declaring a Class**
  - ❖ A class is declared in the same way as a structure is declared.
  - ❖ The keyword class is used to declare a class.
  - ❖ A class declaration specifies the variables and functions that are common to all objects of that class.
  - ❖ The variables declared in a class are known as member variables or data members.
  - ❖ The functions declared in a class are called member functions

Classes

- **Syntax**

```
            class identifier{
             body of the class
              };
```

- **Class :** is a keyword to declare a class
- **identifier>:** is the name of the class

- The class declaration always end with semi colon. All data members and member functions are declared within the braces known as **body** of the class.

Access Specifiers

- The commands that are used to specify the access level of class members are known as access specifiers. Two most important access specifiers are as follows:

❖ **The private Access Specifier**

- It **is** used to restrict the use of class member within the class.
- Any member of the class declared with private access specifier can only be accessed within the class.
- The data members are normally declared with private access specifier. It is because the data of an object is more sensitive.

Access Specifiers

- The commands that are used to specify the access level of class members are known as access specifiers. Two most important access specifiers are as follows:

❖ **The public Access Specifier**

- It is used to allow the user to access a class member within the class as well as outside the class.
- Any member of the class declared with public specifier can be accessed from anywhere in the program.
- The member functions are normally declared with public access specifier. It is because the user access functions of an object from outside the class

Creating Objects

- A class is simply a model or prototype for creating objects. It is like a new data type that contains both data and functions
- An object is created in the same way as other variables are created
- When an object of a class is created, the space for all data members defined in the class is also allocated in the memory according to their data types
- An object is also known as instance of the class. The process of creating an object of a class is also called instantiation.
- Objects are sometimes called instance variables

Creating Objects

- **Syntax**
  - ❖ The syntax of creating an object of a class is also as follows:

    class_name object_name;

- class_name: is the name of the class whose type of object is to be created
- object_name: is the name of the object to be created. The rules for object name are same as the rules for declaring a variable.

- **Example**
  - ❖ **Test obj = new Test();**
  - ❖ The above example declares an object obj of type **Test.** The object contains all data members that are defined in class **Test.**

Executing Member Functions

- An object of a particular class contains all data members as well as member functions defined in that class
- The data members contains the values related to the object
- The member functions are used to manipulate data members
- The member functions can be executed only after creating an object

- The member functions can be executed only after creating an object

Executing Member Functions

- **Syntax**
  - **object_name.function();**
- **object_name**: is the name of the object whose member function is to be executed.
- **function**: is the name of the member function to be executed. Any required parameters are also passed to the member function in parenthesis.

- The object name and member function are separated by **dot operator**

- **Example**
  - **Test obj = new Test();**
  - **obj. input();**

Program Example

- We’ll use objects to represent distances measured in the English system

```
import java.util.Scanner;

class Distance {
    private int feet;
    private float inches;

    // Method to set the distance using arguments
    void setDist(int ft, float in) {
        feet = ft;
        inches = in;
    }

    // Method to get the distance from the user
    void getDist() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter feet: ");
        feet = sc.nextInt();
        System.out.print("Enter inches: ");
        inches = sc.nextFloat();
    }
  // Method to display the distance
    void showDist() {
        System.out.println(feet + "'-" + inches + "\"");
    }
```

```
public static void main(String[] args) {
        Distance dist1 = new Distance();
        Distance dist2 = new Distance();

        dist1.setDist(11, 6.25f); // Set dist1
        dist2.getDist(); // Get dist2 from the user

        // Display the distances
        System.out.print("dist1 = ");
        dist1.showDist();
        System.out.print("dist2 = ");
        dist2.showDist();
    }
}
```

Example

Write a class Marks with three data members to store three marks. Write three member functions in() to input marks, sum() to calculate and return the sum and avg() to calculate and return the average marks.

Example

```
import java.util.Scanner;

class Marks {
    private int a, b, c;

    // Method to take input from the user
    void in() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter three marks: ");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
    }

    // Method to calculate the sum of the marks
    int sum() {
        return a + b + c;
    }

    // Method to calculate the average of the marks
    float avg() {
        return (a + b + c) / 3.0f;
    }

    public static void main(String[] args) {
        Marks m = new Marks();
        int s;
        float a;

        m.in(); // Get marks from the user
        s = m.sum(); // Calculate sum
        a = m.avg(); // Calculate average

        // Display the sum and average
        System.out.println("Sum = " + s);
        System.out.println("Average = " + a);
    }
}
```