Programming with C and C++

CSC-101 (Lecture 31)

Dr. R. Balasubramanian
Professor
Department of Computer Science and Engineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Roorkee
Roorkee 247 667

bala@cs.iitr.ac.in
https://faculty.iitr.ac.in/cs/bala/

- Difference between Static Array and Dynamic Array

Two Dimensional Arrays

- In C++

Static Array: `int  arr_2d[10][12];`

Dynamic Array:

```
  int **A;
A= new int*[rows];
 for (int i=0; i<rows; i++)
  A[i]=new int [cols];
```

- Two-dimensional arrays need not be rectangular. Each row can be a different length. Here's an example:

```
int **A;
A= new int*[rows];
A[0] = new int [1]; // A's first row has 1 column
A[1] = new int [2];  // A's second row has 2 columns
A[2] = new int [3];  // A's third row has 3 columns
A[3] = new int [5];  // A's fourth row has 5 columns
A[4] = new int [5];  // A's fifth row also has 5 columns
```

Classes and Objects in C++

- In object-oriented programming technique, we design a program using objects and classes.
  - object is an entity that has state and behavior. Here, state means data and behavior means functionality.
  - Object is created during run time.

Objects in C++

- Object is an instance of a class. All the members of the class can be accessed through object.

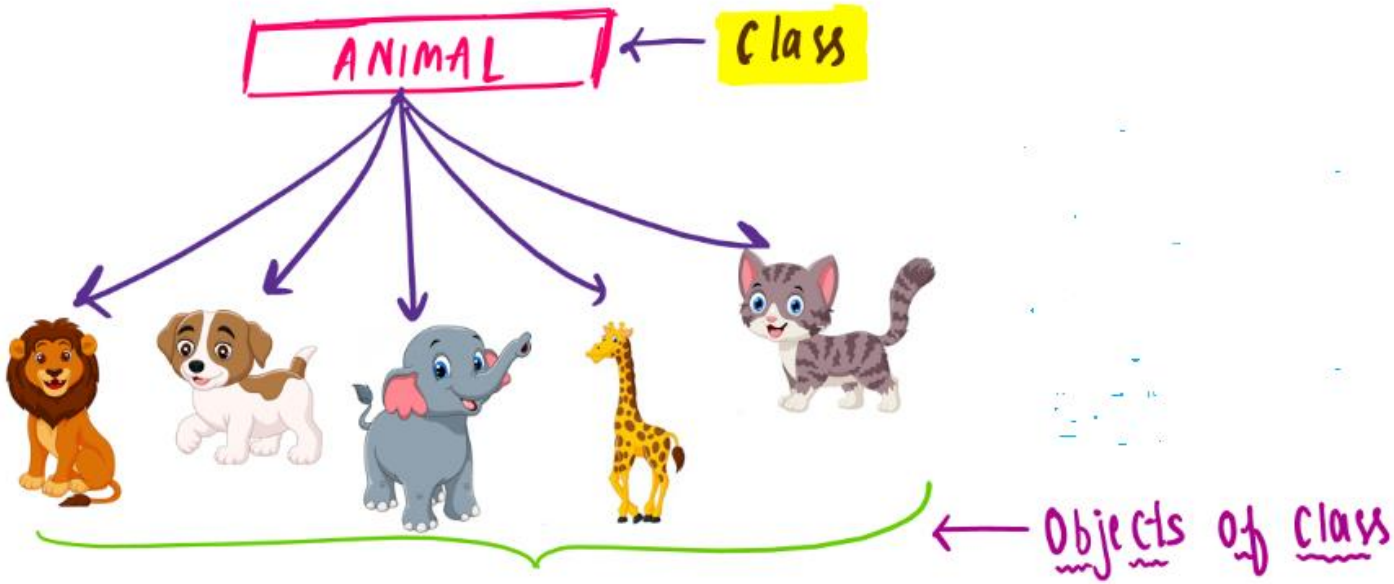
```
Time T1;
```

- Objects can be used to represent physical objects in the world, such as chair, bike, marker, pen, table, car etc.
- Objects can be physical or logical (tangible and intangible).
- Banking system is the example of intangible object.

An object has three characteristics

- **state:** represents data (value) of an object.
- **behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
- **identity:** Object identity is typically implemented via a unique ID.

- For Example: Pen is an object. Its name is Parker, color is Golden etc. known as its state. It is used to write, so writing is its behavior.
- **Object is an instance of a class.** Class is a template or blueprint from which objects are created. So object is the instance(result) of a class.



C++ Class

- Class is a group of similar objects.
- It is a template from which objects are created.
- It can have fields, functions, constructors etc.

A class in C++ can contain:

- Member data
- Member functions
- constructor
- block
- class

C++ Class

```
class <class_name>{
    member data; //field
    member function();
};
```

Example,

```
class Student
 {
     public:
     int id;  //field or data member
     float salary; //field or data member
     String name;//field or data member
 }
```

Class and Objects, Example 1

https://ideone.com/YySMvp

```
#include <iostream>
using namespace std;

class Student {
   public:
      int id;//data member (also instance variable)
      string name;//data member(also instance variable)
};

int main() {
    Student s1; //creating an object of Student
    s1.id = 18;
    s1.name = "Virat Kohli";
    cout<<s1.id<<endl;
    cout<<s1.name<<endl;
    return 0;
}
```

stdout

```
18
Virat Kohli
```

Example 2

```cpp
#include <iostream>

using namespace std;

class Box {
   public:
      double length;   // Length of a box
      double breadth;  // Breadth of a box
      double height;   // Height of a box
};

int main() {
   Box Box1;        // Declare Box1 of type Box
   Box Box2;        // Declare Box2 of type Box
   double volume = 0.0;     // Store the volume of a box here

```

https://ideone.com/B3fnpz

```
    // box 1 specification
    Box1.height = 5.0;
    Box1.length = 6.0;
    Box1.breadth = 7.0;

    // box 2 specification
    Box2.height = 10.0;
    Box2.length = 12.0;
    Box2.breadth = 13.0;

    // volume of box 1
    volume = Box1.height * Box1.length * Box1.breadth;
    cout << "Volume of Box1 : " << volume <<endl;

    // volume of box 2
    volume = Box2.height * Box2.length * Box2.breadth;
    cout << "Volume of Box2 : " << volume <<endl;
    return 0;
}
```

stdout

```
Volume of Box1 : 210
Volume of Box2 : 1560
```

Initialize and Display data through member function

https://ideone.com/SMO06J

```
#include <iostream>
using namespace std;
class Student {
    public:
        int id;//data member (also instance variable)
        string name;//data member(also instance variable)
        void insert(int i, string n)
         {
            id = i;
            name = n;
         }
        void display()
         {
            cout<<id<<"  "<<name<<endl;
         }
};

```

```
int main(void) {
    Student s1; //creating an object of Student
    Student s2; //creating an object of Student
    s1.insert(18, "Virat");
    s2.insert(45, "Rohit");
    s1.display();
    s2.display();
    return 0;
}
```

stdout

```
18   Virat
45   Rohit
```

Storing and displaying employee information using method.

https://ideone.com/TFhgqS

```
#include <iostream>
using namespace std;
class Employee {
   public:
       int id;//data member (also instance variable)
       string name;//data member(also instance variable)
       float salary;
       void insert(int i, string n, float s)
        {
            id = i;
            name = n;
            salary = s;
        }
       void display()
        {
            cout<<id<<"  "<<name<<"  "<<salary<<endl;
        }
};

```

```
19
20 int main(void) {
21     Employee e1; //creating an object of Employee
22     Employee e2; //creating an object of Employee
23     e1.insert(8, "Ravindra",5000000);
24     e2.insert(99, "Ashwin", 7500000);
25     e1.display();
26     e2.display();
27     return 0;
28 }
29
```

Success #stdin #stdout 0.01s 5508KB

```
8   Ravindra   500000
99  Ashwin   750000
```

Example 1

- Design a C++ class for basic geometry calculations. You can have methods for calculating area, perimeter, and other geometric properties of shapes.

https://ideone.com/35HNjD

```cpp
#include <iostream>
#include <cmath>

using namespace std;

class GeometryCalculator {
public:
    // Calculate the area of a rectangle
    static double calculateRectangleArea(double length, double width) {
        return length * width;
    }

    // Calculate the perimeter of a rectangle
    static double calculateRectanglePerimeter(double length, double width) {
        return 2 * (length + width);
    }

```

```
    // Calculate the area of a circle
    static double calculateCircleArea(double radius) {
        return M_PI * radius * radius;
    }

    // Calculate the circumference of a circle
    static double calculateCircleCircumference(double radius) {
        return 2 * M_PI * radius;
    }

    // Calculate the area of a triangle using Heron's formula
    static double calculateTriangleArea(double sideA, double sideB, double sideC) {
        double s = (sideA + sideB + sideC) / 2.0;
        return sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));
    }
};

```

```
int main() {
    double length = 5.0;
    double width = 3.0;
    double radius = 4.0;
    double sideA = 7.0;
    double sideB = 24.0;
    double sideC = 25.0;

    GeometryCalculator G1;


    // Calculate and display rectangle area and perimeter
    cout << "Rectangle Area: " << G1.calculateRectangleArea(length, width) << endl;
    cout << "Rectangle Perimeter: " << G1.calculateRectanglePerimeter(length, width)
    << endl;
```

```
    // Calculate and display circle area and circumference
    cout << "Circle Area: " << G1.calculateCircleArea(radius) << endl;
    cout << "Circle Circumference: " << G1.calculateCircleCircumference(radius)
    << endl;

    // Calculate and display triangle area
    cout << "Triangle Area: " << G1.calculateTriangleArea(sideA, sideB, sideC)
    << endl;

    return 0;
}
```

stdout

```
Rectangle Area: 15
Rectangle Perimeter: 16
Circle Area: 50.2655
Circle Circumference: 25.1327
Triangle Area: 84
```

Example 2

- write a C++ program using class and objects to add two distances given in meter and centimeter

```cpp
#include <iostream>

using namespace std;

class Distance {
private:
    int meters;
    int centimeters;

public:
    void getDistance() {
        cout << "Enter meters: ";
        cin >> meters;
        cout << "Enter centimeters: ";
        cin >> centimeters;
    }

```

https://ideone.com/ECT4fj

```
    void displayDistance() {
        cout << "Distance: " << meters << " meters " << centimeters
        << " centimeters" << endl;
    }

    Distance addDistances(const Distance& d1, const Distance& d2) {
        Distance result;
        result.meters = d1.meters + d2.meters;
        result.centimeters = d1.centimeters + d2.centimeters;

        if (result.centimeters >= 100) {
            result.meters += result.centimeters / 100;
            result.centimeters = result.centimeters % 100;
        }

        return result;
    }
};
```

```
}

int main() {
    Distance distance1, distance2, result;

    cout << "Enter the first distance:" << endl;
    distance1.getDistance();
    cout << "Enter the second distance:" << endl;
    distance2.getDistance();

    result = result.addDistances(distance1, distance2);

    cout << "Sum of the distances:" << endl;
    result.displayDistance();

    return 0;
}
```

stdin

```
10 95
20 85
```

stdout

```
Enter the first distance:
Enter meters: Enter centimeters: Enter the second distance:
Enter meters: Enter centimeters: Sum of the distances:
Distance: 31 meters 80 centimeters
```

Thank You!