SLIIT ACADEMY

Higher Diploma in Information Technology
Year 1, Semester 1

Introduction to Programming(C++)

Lecture 11 : Object Oriented Programming Concepts in C++

Intended Learning Outcomes

End of this lecture you will be able to learn ,

LO1 : Understand the concept of OOP

LO2 : Understand the concept of Class & Object.

LO3 : Identify the different concepts of OOP

What is OOP?

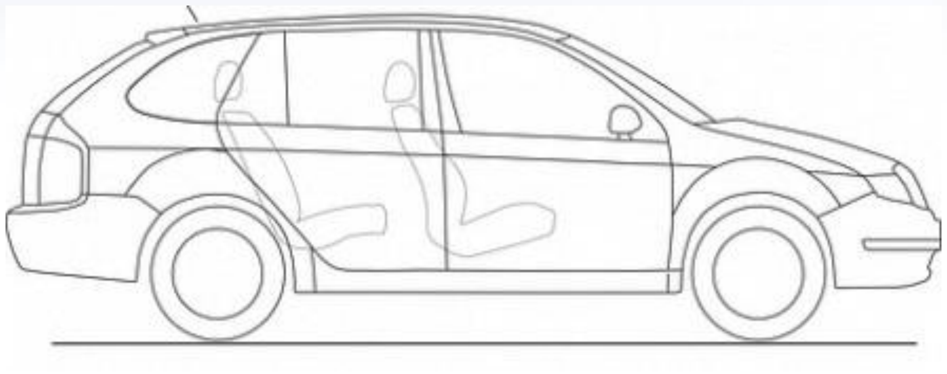
Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

Advantages over procedural programming

- OOP is faster and easier to execute.
- OOP provides a clear structure for the programs.
- OOP helps to keep the C++ code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug.
- OOP makes it possible to create full reusable applications with less code and shorter development time.

What is a Class?

- A class is a blueprint /template for objects.



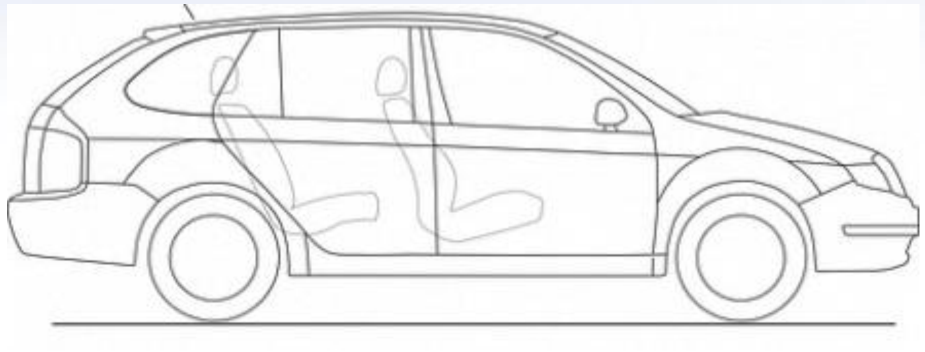
Class : Car

Properties

- brand
- model
- color
- year

What is an Object?

- An object is an instance of a class. Objects have a **state and behavior**.



Class

- brand
- model
- color
- year

Object1

- brand=Mitsubishi
- model =lancer
- color= Perl white
- year = 2019

Object2

- brand=Mercedes-Benz
- model =C
- color= Gold
- year = 2019

C++ Classes/Objects

- C++ is an **object-oriented programming language.**
- C++ is **associated with classes and objects**, along with its attributes and methods.
- Attributes and methods are basically variables and functions that belongs to the class. These are often referred to as **"class members".**
- A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.

Defining a Class

- A class definition begins with the keyword *class*.
- The body of the class is contained within a set of braces, { } ;

```
class <class_name>
{
    <type> <identifier_list>;
    <type> <identifier_list>;
};
```

Should be a valid identifier

Class body contains the members (attributes and methods)

C++ Access Specifiers

- Access specifiers define how the members (attributes and methods) of a class can be accessed.
- In C++, there are three access specifiers:
  - public - members are accessible from outside the class..
  - private - members cannot be accessed (or viewed) from outside the class.
  - protected - members cannot be accessed from outside the class, however, they can be accessed in inherited classes.

C++ Access Specifiers

- Within the body, the keywords private: and public: specify the access level of the members of the class.

    the default is private

- As a good practice attribute are declared in the private: section of the class and the methods are declared in the public: section.

```
class <class_name>
{
private:
    --------------
    --------------
public:
    --------------
    --------------
};
```

Example with Members : Class Circle

```
class Circle
{
     private:
      double radius;
     public:
      double getDiameter();
      double getArea();
      double getCircumference();
};
```

No need for other classes to access and retrieve its attributes directly.

The class methods are responsible for that only.

They are accessible from outside the class, and they can access the members.

Create an Object

- Declaring a variable of a class type creates an object. It is also called Instantiation.
- Once an object of a certain class is instantiated, a new memory location is created.
- To create an object of Class, specify the class name, followed by the object name.

```
<class_name> <obj_name>;
```

Member Functions of Classes in C++

- Methods are functions that belongs to the class.
- There are two ways to define functions that belongs to a class:
  - Inside class definition
  - Outside class definition

Define Member Functions inside the Class

```
class Circle
{
     private:
      double radius;
     public:
       double getArea()
       {
         double area = 3.14 * radius * radius;
         return area;
       }
};
```

Define Member Functions Outside the Class

- To define a function outside the class definition, you must declare it inside the class and then define it outside of the class.
- This is done by specifying the name of the class, followed the **binary scope resolution :: operator**, followed by the name of the function:

Format for defining member functions

```
ReturnType ClassName::MemberFunctionName( )
{
        ............................................
}
```

SLIIT ACADEMY

Define Member Functions Outside the Class

```
class Circle

{

     private:

      double radius;

     public:

      double getCircumference();

};

// Method/function definition outside the class

double  Circle :: getCircumference() {

      double p = 2 * 3.14 * radius ;

      return p;

}
```

C++ Constructors

- A constructor in C++ is a special method that is automatically called when an object of a class is created**(instantiated).**
- To create a constructor, use the same name as the class, followed by parentheses ():
- **Note: The constructor has the same name as the class name, it is always public using to initialize data members and it does not have any return value.**
- Can have several constructors ➔ **Function overloading**

C++ Constructors

```
class Circle
{
     private:
        double radius;
     public:
        Circle();
        Circle(int r);
        double getArea();
        double getCircumference();
};
```

$\rightarrow$ **Constructer with no argument**

$\rightarrow$ **Constructer with one argument**

Constructor Parameters :

Constructors can also take parameters (just like regular functions), which can be useful for setting initial values for attributes.

Accessing Class Members

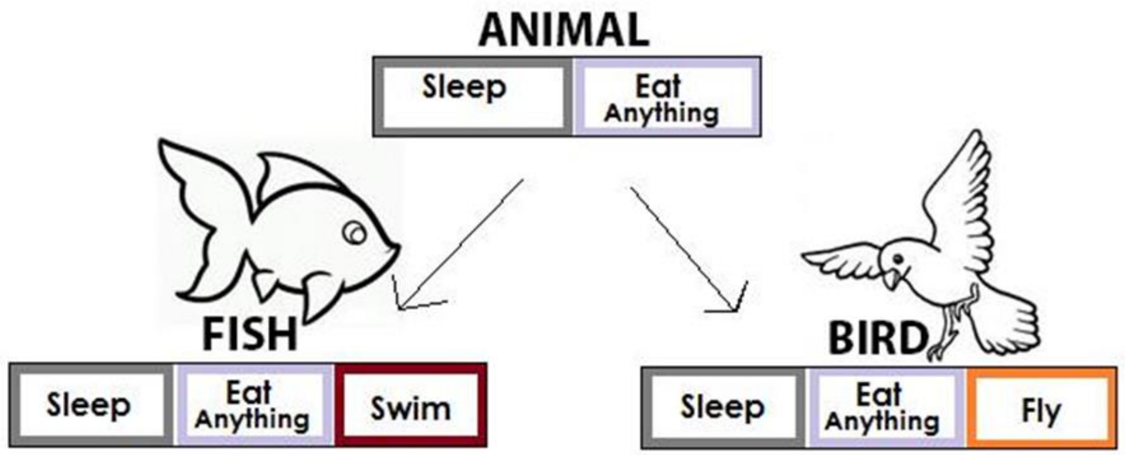
- Operators to access class members
  - Identical to those for **struct**s
  - Dot member selection operator (**.**)
    - Object
    - Reference to object
  - Arrow member selection operator (**->**)
    - Pointers

Encapsulation

- The meaning of Encapsulation, is to make sure that "sensitive" data is hidden from users.
- To achieve this, **you must declare class variables/attributes as private (cannot be accessed from outside the class).**

Inheritance

- Inheritance means that one class inherits the characteristics of another class.



Polymorphism

- Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.
- Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to perform different tasks. This allows us to perform a single action in different ways.

Reasons for OOP

- Simplify programming
- Interfaces

  Information hiding: Implementation details hidden within classes themselves

- Software reuse

  Class objects included as members of other classes

SLIIT ACADEMY

Summary

- What is OOP?
- Advantages of OOP over procedural programming
- What is a Class?
- What is an Object
- Define a Class
- Access modifiers in C++
- Create the object
- Define the member functions in C++
- C++ Constructors
- OOP Concepts
- Reasons for OOP