The Islamic University of Diwaniya

Computer Technique Engineering Department

Object-Orient-Programming

Lecture:5

Date: 25-11-2024

Object – Oriented Programming (OOP) :

Object oriented programming is a fairly new way to approach the task of programming. It supersedes the so called **procedural or structured programming languages** like Pascal or C, that have been around since the 1960s. The essence of structured programming is to **reduce a program into smaller parts** and then code these elements more or less independently from each other. Although structured programming has yielded excellent results when applied to moderately complex programs, it fails when a program reaches a certain size. The **basic idea** was to combine data and the functions that operate on that data into a single unit called an **object** figure(1). In OOP, when you approach a problem you should no longer be thinking about how to divide it into functions. Rather you will be considering what objects to use.

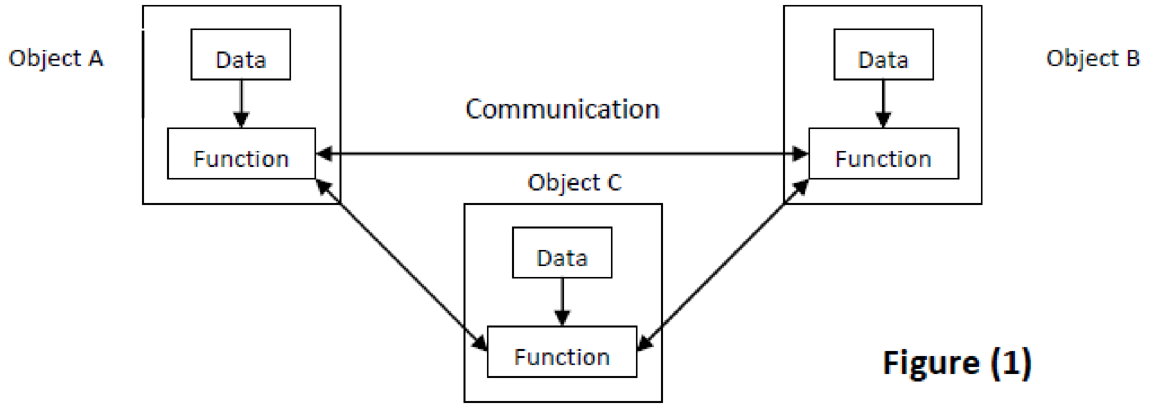

Figure (1)

Features of Object oriented P.

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data structures are designed such that they characterize the objects.
- Functions that operate on the data of an object are tied together in the data structure.
- Data is hidden and cannot be accessed by external functions.
- Objects may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Follows bottom – up approach in program design.

Benefits of OOP:

OOP offers several benefits to both the program designer and the user. And this benefits are:

1- Through inheritance, we can eliminate redundant code and extend the use of existing classes.

2- We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.

3- The principle of data hiding helps the programmer to built secure programs that cannot be invaded by code in other part of the program.

4- It is possible to have multiple instances of an object to co-exist without any interference.

5- It is possible to map objects in the problem domain to those in the program.

6- It is easy to partition the work in project based on objects.

7- The data-centered design approach enables us to capture more details of a model in implementable form.

8- Object-oriented systems can be easily upgraded from small to large systems.

9- Software complexity can be easily managed.

Applications of OOP:

- ➢ Real-time system.
- ➢ Simulation and modeling.
- ➢ Object-oriented databases.
- ➢ Hypertext, hypermedia and expertext.
- ➢ AI and expert systems.
- ➢ Neural networks and parallel programming.
- ➢ Decision support and office automation systems.
- ➢ CIM/CAM/CAD systems.

We shall discuss these concepts

Objects

Objective-C is an object-oriented programming language developers use to create applications for Apple's mac OS, iOS, and iPad operating systems.

Among computer programming languages, Objective-C is easy to understand, and is known for its dynamic binding and reflection capabilities.

Moreover, this programming language can be used with C and C++ code, allowing developers to take advantage of existing libraries and frameworks. It is also commonly used for developing games and other interactive applications.

Class

In C++, a class is a template used to define objects in object-oriented programming (OOP). A class contains attributes (data members) and methods (functions) that represent the data and behaviors.

Defining a class:

```
class ClassName {
public: // Or private, protected
   // Attributes
   // Methods
};
```

Simple Example of a class:

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

class Car {
public:
    string brand;  // Attribute
    int year;      // Attribute

    void display() {  // Method
        cout << "Brand: " << brand << ", Year: " << year << endl;
    }
};

int main() {
    Car myCar; // Create an object of the class
    myCar.brand = "Toyota";
    myCar.year = 2022;

    myCar.display(); // Call the method
    return 0;
}
```

Explanation:

1. Defining the class:

- Car is defined as a class containing attributes (brand and year) and a method (display).

2. Creating an object:

- myCar is an object created from the Car class.
- Access attributes and methods using the object and the dot operator (.).

3. Access Modifiers:

- public: Accessible outside the class.
- private: Not accessible directly from outside the class (default).
- protected: Accessible only via inheritance.

Inheritance Example:

```
class ElectricCar : public Car {
public:
   int batteryLife;
};
```

Benefits of class:

- Organizes code.
- Enhances reusability.
- Supports core OOP principles: Encapsulation, Inheritance, Polymorphism.