Programming in C++

Introduction to C++

- Programming Concept
- Basic C++
- C++ Extension from C

Focus

- Focus on
  - Programming Concepts
  - Programming Design Techniques
- Don’t get lost in
  - Language Technical Details

What is programming?

Programming is taking

A ***problem***

Find the area of a rectangle

A set of ***data***

length

width

A set of ***functions***

area = length * width

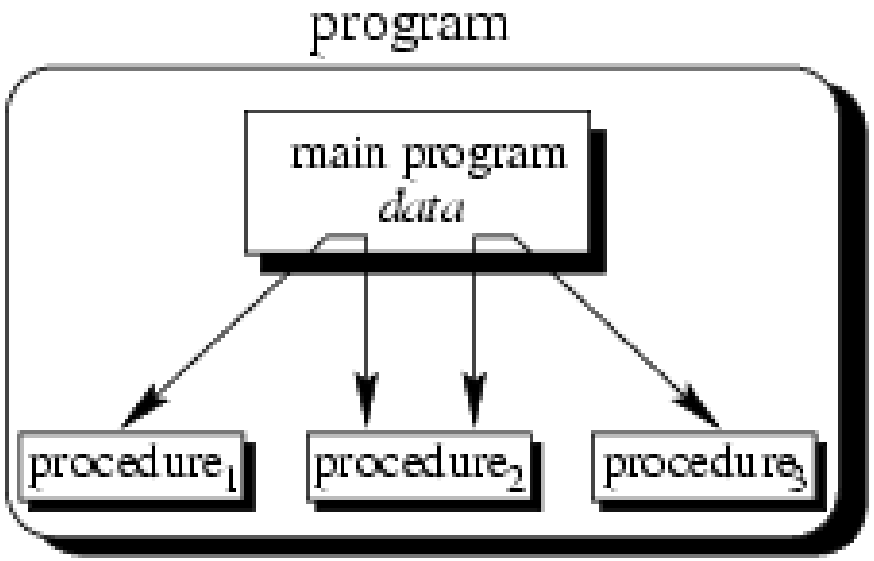
Then,

Applying functions to data to solve the problem

Programming Concept Evolution

- Unstructured
- Procedural
- Object-Oriented

Procedural Concept



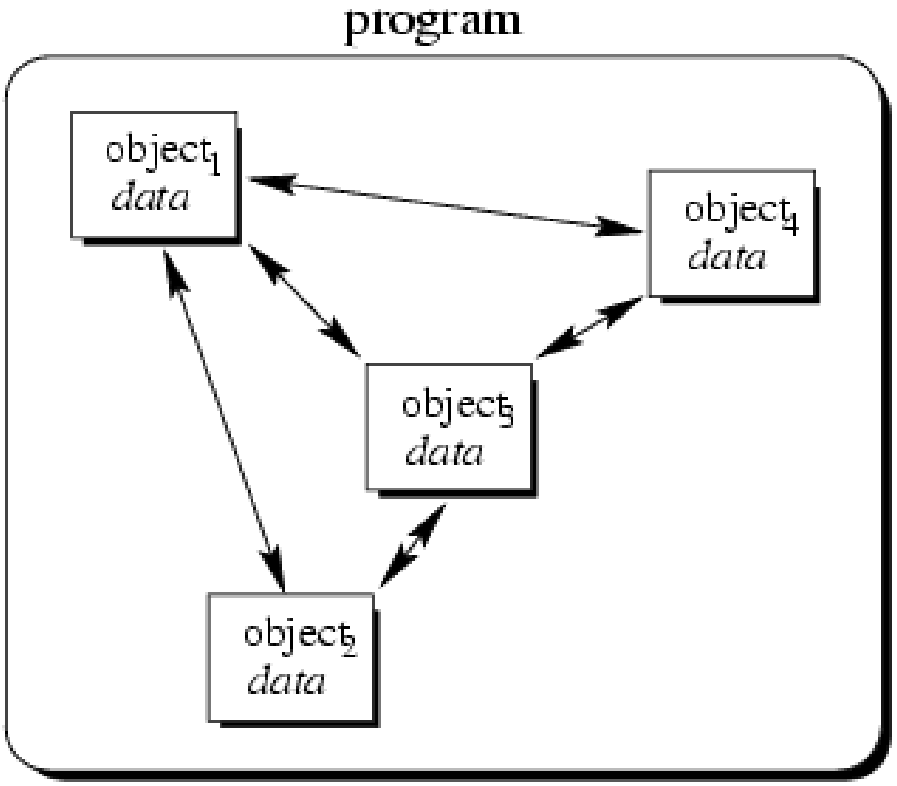
- The main program coordinates calls to procedures and hands over appropriate data as parameters.

Procedural Concept (II)

- Procedural Languages
  - C, Pascal, Basic, Fortran
  - Facilities to
    - Pass arguments to functions
    - Return values from functions
- For the rectangle problem, we develop a function

```
int compute_area (int l, int w){
        return ( l * w );
}
```

Object-Oriented Concept



- Objects of the program interact by sending messages to each other

Objects

An object is an encapsulation of both functions and data

- ***Objects are an Abstraction***
  - *represent real world entities*
  - *Classes are data types that define shared common properties or attributes*
  - *Objects are instances of a class*
- ***Objects have State***
  - *have a value at a particular time*
- ***Objects have Operations***
  - *associated set of operations called methods that describe how to carry out operations*
- ***Objects have Messages***
  - *request an object to carry out one of its operations by sending it a message*
  - *messages are the means by which we exchange data between objects*

OO Perspective

Let's look at the Rectangle through object oriented eyes:

- Define a new type Rectangle (a class)
  - Data
    - width, length
  - Function
    - area()
- Create an instance of the class (an object)
- Request the object for its area

In C++, rather than writing a procedure, we define a class that encapsulates the knowledge necessary to answer the question - here, what is the area of the rectangle.

Example Object Oriented Code

```
class Rectangle
{
    private:
        int width, length;
    public:
        Rectangle(int w, int l)
            {
                width = w;
                length = l;
            }
        int area()
        {
            return width*length;
        }
};
```

```
main()
{
    Rectangle rect(3, 5);
    cout << rect.area()<<endl;
}
```

Object-Oriented Programming Languages

- Characteristics of OOPL:
  - Encapsulation
  - Inheritance
  - Polymorphism
- OOPLs support :
  - Modular Programming
  - Ease of Development
  - Maintainability

Characteristics of OOPL

- **Encapsulation:** Combining data structure with actions
  - Data structure: represents the properties, the state, or characteristics of objects
  - Actions: permissible behaviors that are controlled through the member functions

  Data hiding: Process of making certain data inaccessible
- **Inheritance:** Ability to derive new objects from old ones
  - permits objects of a more specific class to inherit the properties (data) and behaviors (functions) of a more general/base class
  - ability to define a hierarchical relationship between objects
- **Polymorphism:** Ability for different objects to interpret functions differently

REAL TIME EXAMPES OF POPS AND OOPS WITH SOME SITUATION

- Suppose that we suddenly need to add a bus form, that records the following information:

  Color, Engine Size, Transmission Type

  Number of passengers

- *Procedural:* We need to recreate the entire form, repeating the code for Color, Engine Size, and Transmission Type.

- *OOP:* We simply extend the vehicle class with a bus class and add the method, number Of Passengers.

SITUATION 2

- If the client in vehicle company wants to change the colors of all the vehicles for some reason.

- ***Procedural:*** We have to change three different forms: cars, trucks, and buses color differently one by one for the client as its follows procedure step by step .

- ***OOP:*** We change the color method in the vehicle class and because the car, truck, and bus classes all are inherited from one derived class they are automatically updated.

Procedure Oriented Programming

Advantages

- Its relative simplicity, and ease of implementation of compilers and interpreters.
- An easier way to keep track of program flow.
- Needs only less memory.

Disadvantages

- Data is exposed to whole program, so no security for data.
- The does not offer the ability to re-use the same code at different places.
- Difficult to relate with real world objects.
- Difficult to create new data types reduces extensibility.

Fortran

Object Oriented Programming

Advantages

- It is easy to understand.
- It models the real world(i.e) it can be easily extended.
- It offers reusability (i.e) already created classes can be reuse without having to write them again.
- These are easier to test and maintain.

Disadvantages

- It needs a proper planning and proper designing
- Program designing is sometime tricky.

	Procedure Oriented Programming	Object Oriented Programming
Divided Into	In POP, program is divided into small parts called **functions**.	In OOP, program is divided into parts called **objects**.
Importance	In POP,Importance is not given to **data** but to functions as well as **sequence** of actions to be done.	In OOP, Importance is given to the data rather than procedures or functions because it works as a **real world**.
Approach	POP follows **Top Down approach**.	OOP follows **Bottom Up approach**.
Access Specifiers	POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
Expansion	To add new data and function in POP is not so easy.	OOP provides an easy way to add new data and function.
Data Access	In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system.	In OOP, data can not move easily from function to function,it can be kept public or private so we can control the access of data.
Data Hiding	POP does not have any proper way for hiding data so it is **less secure**.	OOP provides Data Hiding so provides **more security**.
Overloading	In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
Examples	Example of POP are : C, VB, FORTRAN, Pascal.	Example of OOP are : C++, JAVA, VB.NET, C#.NET.

Basic C++

- Inherit all ANSI C directives
- Inherit all C functions
- You don’t have to write OOP programming in C++

Basic C++ Extension from C

- comments

```
/* You can still use the old comment style, */
/* but you must be // very careful about mixing them */
// It's best to use this style for 1 line or partial lines
/* And use this style when your comment
consists of multiple lines */
```

- cin and cout (and #include <iostream.h>)

```
cout << "hey";
char name[10];
cin >> name;
cout << "Hey " << name << ", nice name." << endl;
cout << endl;     // print a blank line
```

- declaring variables almost anywhere

```
// declare a variable when you need it
   for (int k = 1; k < 5; k++){
       cout << k;
   }
```

Basic C++ Extension from C (II)

- const
  - In C, #define statement
    - Preprocessor - No type checking.
    - #define n 5
  - In C++, the const specifier
    - Compiler - Type checking is applied
    - const int n = 5; // declare and initialize
- New data type
  - Reference data type “&”.

```
int ix; /* ix is "real" variable */
int & rx = ix; /* rx is “alias” for ix. Must initialize*/
ix = 1; /* also rx == 1 */
rx = 2; /* also ix == 2 */
```

C++ - Advance Extension

- C++ allows function overloading
  - In C++, functions can use the same names, within the same scope, if each can be distinguished by its name *and* signature
  - The signature specifies the number, type, and order of the parameters expressed as a comma separated list of argument types

C++

- Is a better C
- Expressive
- Supports Data Abstraction
- Supports OOP
- Supports Generic Programming
  - Containers
    - Stack of char, int, double etc
  - Generic Algorithms
    - sort(), copy(), search() any container Stack/Vector/List

Take Home Message

- There are many different kinds of programming paradigms, OOP is one among them.
- In OOP, programmers see the execution of the program as a collection of dialoging objects.
- The main characteristics of OOPL include encapsulation, inheritance, and polymorphism.