Classes and Objects in C++ programming language

"Classes, Objects, Functions, Data Hiding"

Fundamentals of OOPs

Shakirullah Waseeb
shakir.waseeb@gmail.com

Nangarhar University

October 3, 2017

Agenda

1. Introduction
2. A simple Class example
3. Description of the example
4. Working on our own use cases
5. What's next?
6. Questions and Discussion

Introduction

- So for we studied functions, structures, and other program constructs
- Finally, we are ready to start programming in classes and objects way
- In this lecture we will start with simple classes and will switch to complex example
- Let us begin with our first C++ program in the next slide


A Simple Class Prgram

simple.cpp program

```
#include <iostream >\\preprocessor directive
using namespace std;

class simple {
    private:
        int memb;
    public:
        void  set() {
            cout << "Enter an integer value" << endl;
            cin>> memb;
        }
        void  print() {
            cout << "The member has value: " << memb;
        }
};
```

Program Continue - - -

program continue

```
void main() {
    simple simp1, simp2;
    simp1.set();
    simp2.set();
    simp1.print();
    simp2.print();
}
```


Class declaration and definition

- Class syntax: class body is delimited by {} followed by ;
- Class members: class simple contains two types of members
  Member *data* : memb
  Member *functions* : set() and print()
- Class data hiding: Has two access specifiers:
  public : can be accessed outside the class
  private : can't be accessed outside the class
- Functions defined inside the class are similar to inline functions, while defined outside the class are not normally inline


Object instantiation and usage

- Objects : has same **relationships** to a **class** likewise a *variable* has to a *data type*
- an object is said to be instance of a given class
- In example program the simple class is defined outside the main method, then simp1 and sim2 are defined as objects of give class in the main method
- The process of object creation is called instantiation and objects are sometimes called instance variables
- Space is set aside for each object during instantiation


Object instantiation and usage - - - continue

- Public members of the class can be accessed via its objects
- To access a member the dot (.) operator connects the object and the members
- Accessing a member data is similar to that of accessing structure
- Calling a member method is connect the function call signature with the dot operator and object
- Some object-oriented languages refers to function call by objects as messaging


Sale Purchase Use case

- Item
  data {Id, name, type, unit, price_per_unit, quantity_in_hand},
  functions {new_item, edit_item, change_price, update_quantity}
- Customer
  data {Id, name, DOB, address, contact}
- Sale
  data {Id, date, customer_idm, sale_total, discount}
- SaleDetails
  data {Id, sale_id, item_id}


Cricket Use case

- Player
  data {Id, name, DOB, address, contact, type, playing_style },
  functions {new_player, get_age, change_address}
- Team
  data {Id, name, type, ball_coach_id, bat_coach_id, field_coach_id}
- Match
  data {Id, team_a_id, team_b_id, win_team_id, match_status}
- Stadium
  data {Id, }


What's next?

Dealing with Classes and Objects in C++ programming language

Your Turn: Time to hear from you!

1

[1] https://fensafitters.files.wordpress.com/2013/07/3d095.jpg

References

Robert Lafore
Object-Oriented Programming in C++, 4th Edition .
2002.

Piyush Kumar
Object oriented Programming (Using C++)
http://www.compgeom.com/ piyush/teach/3330