**EEL 2880: Software Techniques: Modified 11 01 2019 Fall 2019**

**Dr. Subbarao V Wunnava: Ref: Chps 15 & 16: C++ Programs**

The uniqueness of C++ platform lies in the classes and Object orientation. With proper declaration, C functions could also be used in the C++ software development.

**CLASSES:**

Classes are an expanded concept of *data structures*: like data structures, they can contain data members, but they can also contain functions as members. They also contain objects on which the member functions operate.  
  
An object is an instantiation of a class. In terms of variables, a class would be the type, and an object would be the variable. Classes are defined using either keyword class

Where class\_name is a valid identifier for the class, object\_names is an optional list of names for objects of this class. The body of the declaration can contain *members*, which can either be data or function declarations, and optionally *access specifiers*.  
  
The format is the same as for plain *data structures*, except that they can also include functions, and have this new things called access specifiers. An *access specifier* is one of the following three keywords: private, public or protected. These specifiers modify the access rights for the members that follow them:

* private members of a class are accessible only from within other members of the same class (or from their *"friends"*).
* protected members are accessible from other members of the same class (or from their *"friends"*), but also from members of their derived classes.
* Finally, public members are accessible from anywhere where the object is visible.  
  By default, all members of a class declared with the class keyword have private access for all its members. Therefore, any member that is declared before any other *access specifier* has private access automatically
* Objects:
* The prime purpose of C++ programming was to add object orientation to the C programming language, which is in itself one of the most powerful programming languages.
* The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods. While designing C++ modules, we try to see whole world in the form of objects. For example a car is an object which has certain properties such as color, number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.
* There are a few principle concepts that form the foundation of object-oriented programming:
* This is the basic unit of object oriented programming. That is both data and function that operate on data are bundled as a unit called as object.

**EEL 2880: Software Techniques: Modified 11 01 2019 Fall 2019**

**Dr. Subbarao V Wunnava: Ref: Chps 15 & 16: C++ Programs**

// 032813\_Helloworld\_C++.cpp : main project file. Inline functions

// EEL 2880 Software Subbarao Spring 2018: Modified Fall 2019

// using inline functions in C++

#include "stdafx.h"

#include "iostream"

#include <stdio.h>

using std::cout;

using std::cin;

using std::endl;

using namespace System;

// in line function in C++ plugs in the code when the function is called

// so that the stack and other overhead are saved and processing speed increases

inline double cube (const double side) // this function code is plugged in when function is called

{

return side\*side\*side;

} //cube calculated and function ended

int main(array<System::String ^> ^args)

{

double sideValue; //value for calculating cube

cout << endl;

Console::WriteLine(" Hellow World: EEL 2880 FIU Spring 2013");

printf ("\a\a\n\n" );

Console::WriteLine(" Subbarao Wunnava: 03 28 2013:");

printf ("\a\a\n\n" );

for (int i = 1; i <=3; i++)

{

cout << ("\a\n enter side value for the cube:: ");

cin >> sideValue; // input side value

cout << (" \a\n Volume of cube with sideValue: ")

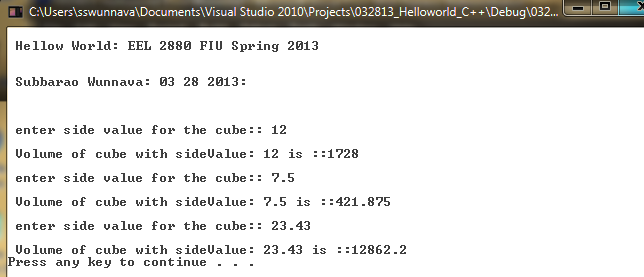
<< sideValue << (" is ::") << cube (sideValue) << endl;

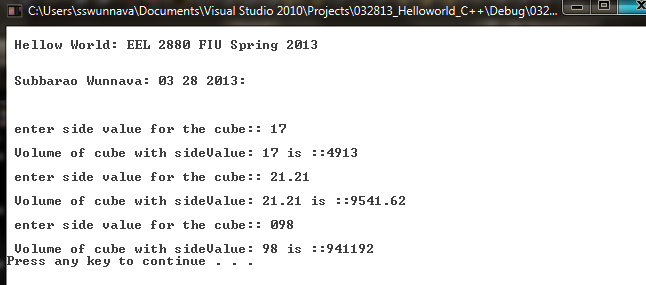
} // end for loop for calculating and printing cube

system ("PAUSE");

return 0;

} // end main





**overloading\_C++\_functions Modified Fall 2019**

**EEL 2880: Subbarao Wunnava: 11 01 2019**

// 11 01 2019\_overloading\_C++\_functions.cpp : main project file.

// 041315\_overloading\_C++\_functions.cpp : main project file.

// Subbarao Wunnava EEL 2880 Fall 2016; Modified 11 01 2018

#include "stdafx.h"

#include <iostream>

#include <stdio.h>

using std::cout; // standardizing cout output function

using std::cin; // standardizing cin input function

using std::endl; // standardizing end line function

using namespace System;

//defining square and quad functions

int square (int x)

{

cout << ("\a\a\n\n square of integer ") << x << " is :" ;

return (x\*x);

} //end function 'square' with integer values

double square (double y)

{

cout << ("\a\a\n\n square of double ") << y << " is :" ;

return (y\*y);

} //end function 'square' with decimal double values

int main( ) // testing the overloading properties in C++

{

printf ("\a\n 040113\_overloading\_C++\_functions.cpp \a\n");

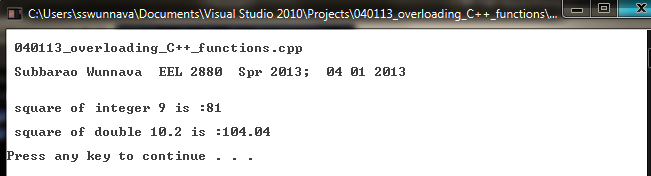
printf (" \a\n Subbarao Wunnava EEL 2880 Spr 2013; 04 01 2013 \a\a\n");

cout << square (9);

cout << square (10.2) << endl << endl;

system ("PAUSE");

} //end main



**EEL 2880 Software Techniques: Modified Fall 2018: Modified 11 01 2019**

**Dr. Subbarao V Wunnava: Classes and Objects**

**C++ Object Oriented Programming examples Chp 16 -19**

**GENERATING HEADER FILES IN C++**

// 041613\_cpp\_gradebook\_class\_cp16.cpp : main project file. Modified 04 04 2015

// Deitel/Subbarao EEL 2880 11 12 2013 16.5: Object Oriented C++ Programs

//Functions operate on specified objects in the Object Oriented Programming

#include "stdafx.h"

#include <iostream>

#include <string>

#include <stdio.h>

using namespace std;

**class GradeBook** //GradeBook class declared

{

public: // these member functions are public and can be used by others

void setCourseName(string name) // member function to set the course name

{

courseName = name; // coursrName is a data member

}

string getCourseName() // member function to get course name

{

return courseName;

}

void displayMessage() // member function to display objected oriented message

{

cout << "Welcome to gradebook for:\t\a" << getCourseName() << "\a\n\a\n" << endl;

}

private: // private declaration of data member CourseName as a string

string courseName;

};// end class GradeBook

**int main()**

{

string nameOfCourse; // string space to store the name of the course

GradeBook myGradeBook; //my GradeBook ojbect created

cout << ("\a\n Deitel/Subbarao EEL 2880 04 16 2013 16.5 modified \n ");

cout << (" \t\t\a : Object Oriented C++ Programs: Chp 16-19: \n ");

cout << myGradeBook.getCourseName() << endl;// Object oriented operation to . get course name

cout << "\a\n enter course name: " << endl;

getline (cin, nameOfCourse);// stores entered course name in nameOfCourse

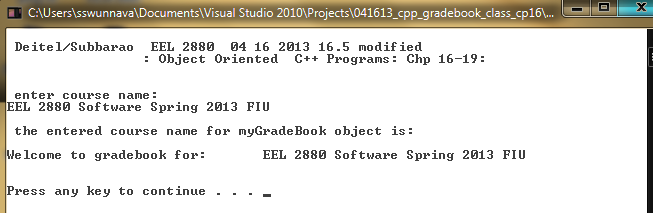
myGradeBook.setCourseName (nameOfCourse);// set the course name for the . myGradeBook object

cout << ("\a\n the entered course name for myGradeBook object is: \a\t\n ") . << endl;

myGradeBook.displayMessage(); // object oriented display

system ("PAUSE");

} //end main program



**Linking two or more files: 11 07 2019: Copy right extended to End 2019**

**GradeBook Class is put in separate header file GradeBook.h**

**Main C++ program put in GradeBook.cpp file, which includes GradeBook.h header file**

**GradeBook. H header file**

// 041613\_cpp\_gradebook\_class\_cp16.cpp : main project file. GRADEBOOK HEADER FILE

// Deitel/Subbarao EEL 2880 04 16 2013 16.5 modified: Object Oriented C++ Programs

// GradeBook class as put as seperate file GradeBook.h

#include "stdafx.h"

#include <iostream>

#include <string>

#include <stdio.h>

using namespace std;

class GradeBook //GradeBook class declared

{

public: // these member functions are public and can be used by others

void setCourseName(string name) // member function to set the course name

{

courseName = name;

}

string getCourseName() // member function to get course name

{

return courseName;

}

void displayMessage() // member function to display objected oriented message

{

cout << "Welcome to gradebook for:\t\a" << getCourseName() << "\a\n\a\n" << endl;

}

private: // private declaration of data member CourseName as a string

string courseName;

};// end class GradeBook

**GradeBook.cpp Source File**

// 041613\_cpp\_gradebook\_class\_cp16.cpp : main project file.

// Deitel/Subbarao EEL 2880 04 16 2013 16.5 modified: Object Oriented C++ Programs

// GradeBook class defined in GradeBook.h header file

// GradeBook main program defined in GradeBook.cpp

#include "stdafx.h"

#include <iostream>

#include <string>

#include <stdio.h>

#include "GradeBook.h"

using namespace std;

int main()

{

printf ("\a\a\n GradeBook Class put in seperate header file GradeBook.h:\a\n");

printf ("\a\a\n main program put in seperate C++ source file GradeBook.cpp:\a\n");

printf ("\a\a\n main C++ source file & header files should be in same directory :\a\n\n");

string nameOfCourse; // string space to store the name of the course

GradeBook myGradeBook; //my GradeBook ojbect created

cout << ("\a\n Deitel/Subbarao EEL 2880 04 16 2013 16.5 modified \n ");

cout << (" \t\t\a : Object Oriented C++ Programs: Chp 16-19: \n ");

cout << myGradeBook.getCourseName() << endl;// Object oriented operation to get course name

cout << "\a\n enter course name: " << endl;

getline (cin, nameOfCourse);// stores entered course name in nameOfCourse string

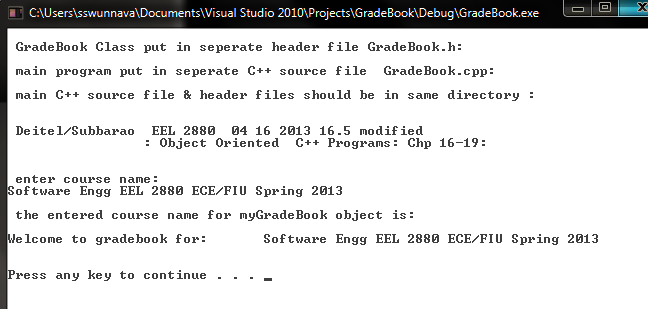
myGradeBook.setCourseName (nameOfCourse);// set the course name for the myGradeBook object

cout << ("\a\n the entered course name for myGradeBook object is: \a\t\n ") << endl;

myGradeBook.displayMessage(); // object oriented display

system ("PAUSE");

} //end main program



// 041613\_cpp\_gradebook\_class\_cp16.cpp : main project file.

// Deitel/Subbarao EEL 2880 04 16 2013 16.5 modified: Object Oriented C++ Programs

#include "stdafx.h"

#include <iostream>

#include <string>

#include <stdio.h>

using namespace std;

class GradeBook //GradeBook class declared

{

public: // these member functions are public and can be used by others

void setCourseName(string name) // member function to set the course name

{

courseName = name;

}

string getCourseName() // member function to get course name

{

return courseName;

}

void displayMessage() // member function to display objected oriented message

{

cout << "Welcome to gradebook for:\t\a" << getCourseName() << "\a\n\a\n" << endl;

}

private: // private declaration of data member CourseName as a string

string courseName;

};// end class GradeBook

int main()

{

string nameOfCourse; // string space to store the name of the course

GradeBook myGradeBook; //my GradeBook ojbect created

cout << ("\a\n Deitel/Subbarao EEL 2880 04 16 2013 16.5 modified \n ");

cout << (" \t\t\a : Object Oriented C++ Programs: Chp 16-19: \n ");

cout << myGradeBook.getCourseName() << endl;// Object oriented operation to get course name

cout << "\a\n enter course name: " << endl;

getline (cin, nameOfCourse);// stores entered course name in nameOfCourse string

myGradeBook.setCourseName (nameOfCourse);// set the course name for the myGradeBook object

cout << ("\a\n the entered course name for myGradeBook object is: \a\t\n ") << endl;

myGradeBook.displayMessage(); // object oriented display

system ("PAUSE");

} //end main program