

Lab # 1

Object:

Introduction to Object Oriented Programming with C++, .Net Framework and Visual Studio 2010.

Theory:

Introduction.

The fundamental idea behind object-oriented languages is to combine into a single unit both *data* and the *functions that operate on that data*. Such a unit is called an *object*. An object's functions, called *member functions* in C++, typically provide the only way to access its data.

To read a data item in an object, member function in the object is called, which in turn will access the data and return the value. The data cannot be accessed directly. The data is *hidden*, so it is safe from accidental alteration. *Data encapsulation* and *data hiding* are key terms in object-oriented languages. This simplifies writing, debugging, and maintaining the program.

A C++ program typically consists of a number of objects, which communicate with each other by calling one another's member functions. Member functions are also called *methods* in some other object-oriented (OO) languages. Also, data items are referred to as *attributes* or *instance variables*.

Characteristics of Object Oriented Languages.

The major elements of object-oriented languages in general, and C++ in particular are:

- Objects
- Classes
- Inheritance.
- Reusability
- Polymorphism and Overloading.

C++ and C.

C++ is derived from the C language. Almost every correct statement in C is also a correct statement in C++, although the reverse is not true. The most important elements added to C to create C++ are concerned with classes, objects and Object-Oriented Programming. C++ programs differ from C programs in other ways, including how I/O is performed and what headers are included. Also, most C++ programs share a set of common traits that clearly identify them as C++ programs. Before moving on to C++'s object-oriented constructs, an understanding of the fundamental elements of a C++ program is required.

The New C++ Headers

When you use a library function in a program, you must include its header. This is done using the `#include` statement. Standard C++ still supports C-style headers for header files that you create and for backward compatibility. However, Standard C++ created a new kind of header that is used by the Standard C++ library.

The new-style C++ headers are an abstraction that simply guarantee that the appropriate prototypes and definitions required by the C++ library have been declared. Since the new-style headers are not filenames, they do not have a .h extension. They consist solely of the header name contained between angle brackets. For example, here are some of the new-style headers supported by Standard C++.

`<iostream> <fstream> <vector> <string>`

Namespaces

When you include a new-style header in your program, the contents of that header are contained in the std namespace. A namespace is simply a declarative region. The purpose of a namespace is to localize the names of identifiers to avoid name collisions. Elements declared in one namespace are separate from elements declared in another. Originally, the names of the C++ library functions, etc., were simply put into the global namespace (as they are in C). However, with the advent of the new-style headers, the contents of these headers were placed in the std namespace.

I/O Operators

`cout << "This is output. \n";`

This statement introduces new C++ features, `cout` and `<<`. The identifier `cout` is a predefined object that represents output stream in C++. The standard output stream represents the screen. The operator `<<` is called the insertion operator. It inserts (or sends) the contents of the variable on its right to the object on its left.

`cin << num1;`

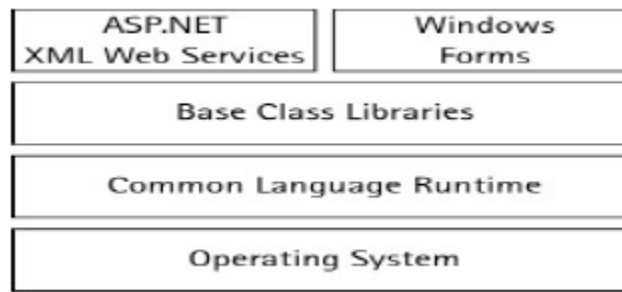
The statement is an input statement and causes program to wait for the user to type in number. The identifier `cin` is a predefined object in C++ that corresponds to the standard input stream. Here stream represents the keyboard. The operator `>>` is known as extraction or get from operator. In general, you can use `cin >>` to input a variable of any of the basic data types plus strings.

A Basic C++ Program.

```
#include <iostream>
using namespace std;
void main()
{
    char course_code[10];
    cout<<"Welcome to OOP for Engineers"<<endl;
    cout<<"Input the course code for this lab:"<<endl;
    cin>>course_code;
    cout<<"The course code for this lab is:"<<course_code<<endl;
}
```

Introduction to dot net framework

.NET is a collection of tools, technologies, and languages that all work together in a framework to provide the solutions that are needed to easily build and deploy truly robust enterprise applications. These .NET applications are also able to easily communicate with one another and provide information and application logic, regardless of platforms and languages.



The first thing that you should notice when looking at this diagram is that the .NET Framework sits on top of the operating system. Presently, the operating systems that can take the .NET Framework include Windows XP, Windows 2000, and Windows NT. There has also been a lot of talk about .NET being ported over by some third-party companies so that a majority of the .NET Framework could run on other platforms as well.

At the base of the .NET Framework is the Common Language Runtime (CLR). The CLR is the engine that manages the execution of the code.

The next layer up is the .NET Framework Base Classes. This layer contains classes, value types, and interfaces that you will use often in your development process. Most notably within the .NET Framework Base Classes is ADO.NET, which provides access to and management of data.

The third layer of the framework is ASP.NET and Windows Forms. ASP.NET should not be viewed as the next version of Active Server Pages after ASP 3.0, but as a dramatically new shift in Web application development. Using ASP.NET, it's now possible to build robust Web applications that are even more functional than Win32 applications of the past.

The second part of the top layer of the .NET Framework is the Windows Forms section. This is where you can build the traditional executable applications that you built with Visual Basic 6.0 in the past. There are some new features here as well, such as a new drawing class and the capability to program these applications in any of the available .NET languages.

Introduction to visual studio 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft for creating, documenting, and debugging programs written in a variety of .net programming languages.

It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Programming Exercise:

Use setw(width) manipulator for formatting output where required.

Q #1: Write a program that read two integers from the keyboard and print their sum and average.

Sample run:

```
Enter first number: 25
Enter second number: 25
Sum is 50
Average is 25
```

Q #2: Write a program that prompts for a person's height in inches. Convert this height measurement into feet by using the conversion factor of `foot2Inch= 12 inch`. Now, the value obtained can easily, be translated into feet and inches which are then output by the program.

Sample run:

```
Enter the height in inches: 20"
This is equivalent to 1' 8"
```

Q #3: Write a program that prompts for time in seconds and output that time in hours, minutes, and seconds. Here student will learn the usage of divide and modulus arithmetic operators in integers.

Sample run:

```
Enter the time in seconds:3713
Hours in time is: 1
Minutes in time: 1
Seconds in time is: 53
```

Q #4: Write a program that prompts for amount in rupees and show how many 1000's, 500's, 100's, 50's, 10's, 5's, 2's and 1's in it.

Sample run:

```
Enter amount in rupees: 5788
1000's in the given amount is: 5
500's in the given amount is: 1
100's in the given amount is: 2
50's in the given amount is: 1
10's in the given amount is: 3
5's in the given amount is: 1
2's in the given amount is: 2
```

1's in the given amount is: 1

Q #5: Write a program that calculates the temperature in Fahrenheit. For that it prompts for temperature in Celsius degrees. Formula to calculate Fahrenheit temperature is $Fahrenheit = Celsius \times (9/5) + 32$. Once if the task done do the vice versa i.e. $Celsius = 5/9(Fahrenheit - 32)$

Sample run:

Enter the Fahrenheit temperature: 98.6
Celsius Temperature: 37.0

Q #6: Write a program that inputs a two digit integer value, and output its reverse order.

Sample run:

Enter a 2 digit integer value: 45
Reverse of this value is: 54

Q #7: Write a program that reads the two digit number as two characters chTen and chUnit and convert that two digit number into an integer value. In order to compute the corresponding integer value, each character must be converted to the digit in the range 0 to 9. this is done by subtracting 48('0') from the ASCII value of the character.

ValueTen=chTen-'0'; // '8'-'0' is 8

ValueUnit=chUnit-'0'; // '2'-'0' is 2

To create integer value from m, the positional value of each digit must be used. In this case multiply ValueTen by 10.

$M = ValueTen \times 10 + ValueUnit; // m = 8 \times 10 + 2 = 82$

Sample run:

Enter a two digit number: 82
Numeric Value is: 82