SECTION 1 C + + PROGRAMMING LAB

Structure **Page Nos.**

1.0 INTRODUCTION

Our attempt in this section is to make you familiar with object oriented programming (OOP) using C + + . We will discuss about how to write and run C + + program. It is essential for you to practice as much as possible for getting more and more understanding in the programming and problem solving. So far you have learnt basics of problem solving and data handling concepts in courses MCS-011 & MCS-023 and their respective lab work. By now you must have experienced that by experimenting with the tool/package/programming language, you gain valuable experience and your efficiency has increased.

By now, you have obtained the practical skills of c programming languages. In C + + , almost the same kind of conditional and logical statements are used as you have used in c programming. To begin with you may consider the C + + programming as a extention of c programming. To become a good C + + **programmer**, it is essential to take one **step at a time**, while programming/ problem solving. Hence, during the problem solving/ programming process, you need to have good understanding of following:

- the tools you are using for program writing and running,
- basic language feature like data types, variables, loops and functions etc.,
- understanding of problem and deciding how to go from an idea to code, and
- a clear, understanding of OOP features need to be used in specific program.

In this lab course, we have used Dev-C + + , for writing and running the programs. The Dev-C + + , is developed by Bloodshed Software. It is a full featured graphical IDE (Integrated Development Environment), which is able to create Windows or console-based C/C + + programs using the MinGW compiler system. We have also added practice problems in this section, which you can attempt and ***enjoy C + + programming***.

1.1 OBJECTIVES

By the end of the practical sessions of this section, you will be able to:

- use basic constructs of C + + in your program;

- define and use objects in C + + programs;
- define and use constructors and destructor in C + + programs;
- use concept of inheritance and polymorphism in C + + programs;
- write programs for I/O handling in C + +;
- create your own C + + templets; and
- handle exceptions in C ++ programs.

1.2 GENERAL GUIDELINES

- You should come prepared for you lab session to properly utilize ther maximum time of Lab session.
- You should attempt all lab exercises/assignments given in the list (session wise).
- You may seek assistance in doing the lab exercises from the available lab instructor.
- For each program, you should add general comments such as your name date of programming etc.
- There should be proper comments for description of the problem, requirement of class, function etc. Proper comments are to be provided as and when necessary in the programming.
- Your program should be interactive and properly documented with real Input/ Output data.
- Proper management of file of Lab record is necessary.Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, program code along with comments and output for various inputs given.
- There are total 10 lab sessions (of 3 hours each) for this lab course.

1.3 SOME C + + COMPILERS

For writing C + + program, you are free to use any C + + compiler and editor available at your centre/home. If you get opportunity you may experiment with more than one C + + compilers. In your c programming lab, you might have used Borland C/C + + compiler. The use of Borland C/C + + compiler is explained to you in BCSL-021 course. List of some compilers which are avauilable for free download is given below. You are advised to check their conditions/licenses before using them.

- Apple C + + .
- Bloodshed Dev-C + + .
- Borland C + +
- Clang C + + ..
- Cygwin (GNU C + +)
- Digital Mars C + +
- DJ Delorie's C + + development system for DOS/Windows (GNU C + +)
- IBM C + + for IBM power, System Z, Bluegene, and Cell.
- Intel C + + for non-commercial development Sun Studio.

Here you are re-introduced about working with Borland C/C + + compiler. For more details you may refer to your BCSL-021 course.

Using Borland C/C + + Compiler

You may use the Borland c/ C + + compiler during the lab sessions under MS-DOS connecting through Windows. A C + + program written in Borland C + + is saved with .cpp extention.

To start Borland C/C + +

Click the **Start** button in the bottom left hand corner of the screen. The **Start** menu pops up. Select **Programs** from the **Start** menu. Select **Borland C/C + + from** the **Programs** menu. Select **Borland C/C + +** from the **Borland C + + menu**. In summary the steps to launching **Borland C/C + +** are:

Start--->Programs--->Borland C + + --->Borland C + +

You should now proceed to the main window for the C/C + + development environment.

Editing and Running C + + Program

You can create a program by entering text that corresponds to C + + statements into a file.The compilatiuon end running process is same as explained in BCSL-021 course. It is no note that Borland compilers for C/C + + runs on DOS prompt (Non-window). Borland compiler for C/C + + is not having good interface compared to Dev C + + IDE. The Dev C + + IDE run on windows and easy to use as it provides very good GUI.

1.4 DEV C + + IDE

Dev-C + + , is a *fully featured easy to use* graphical IDE (Integrated Development Environment), which is able to create Windows or console-based C/C + + programs using the MinGW compiler system. MinGW (Minimalist GNU for Windows) uses GCC (the GNU g++ compiler collection), which is essentially the same compiler system that is in Cygwin (the unix environment program for Windows) and most versions of Linux. The GNU General Public License (GPL) is a free, copyleft license for software.

The Dev-C + + as available as free software (under GPL). Link to Bloodshed Dev-C + + sites is : http://www.bloodshed.net/devcpp.html.

You may download Dev-C + + and should installer put Dev-C + + in the default directory of C:\Dev-Cpp. Hence, later on installation of add-ons or upgrades can be easily done.

Dev C + + Installation:

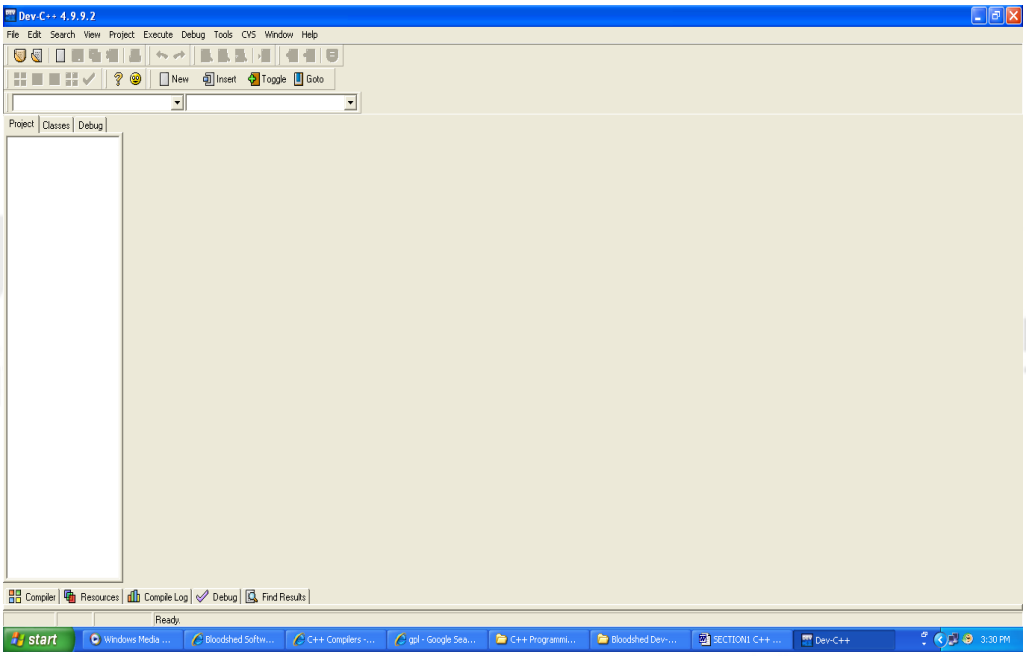
To install Dev-C + + onto your computer , first download the free installation program, copying it to a folder on youre computer.

1. Then install Dev-C + + onto yopur omputer by following the steps:

   - Run the setup program.
   - BE SURE to use the default directory for installation (C:\dev-cpp).

- If you insist on using a different directory, BE SURE to use a directory path that DOES NOT CONTAIN SPACES in the pathname (this means you can't use a subdirectory of C:\Program Files). Otherwise, your good C + + programs will not compile or run and you will wonder why. Spaces in pathnames to your project and source code files don't seem to matter.

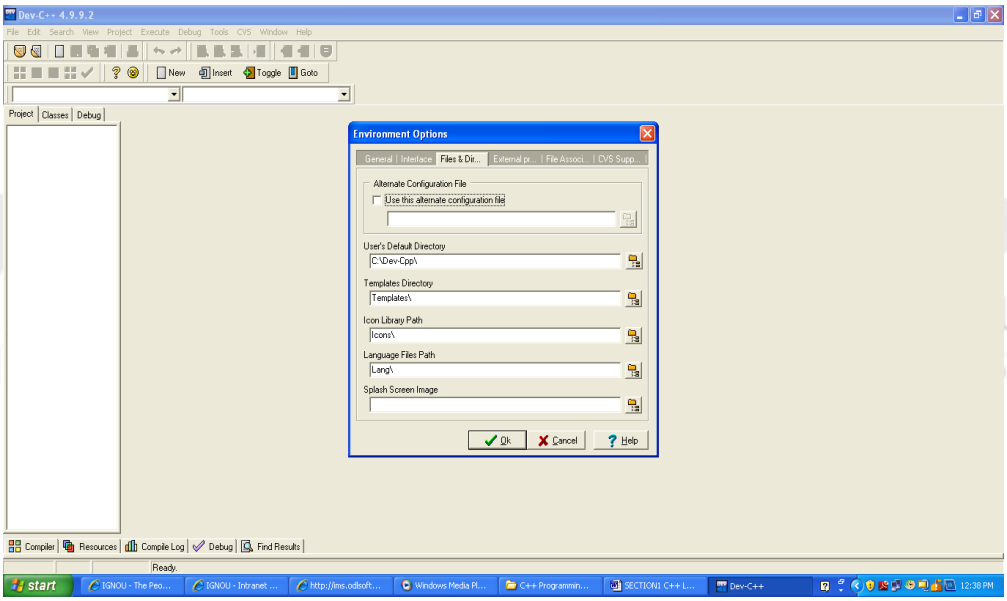
- Run Dev-C + + , you will get following window:



- When first time you run Dev-C + + , it will ask you for your language.
- Chose English as language.

2. Now you may set the default user directory for saving/keeping your C + + programs.

- Under Tools | Environment Options, click the Files & Directories tab.

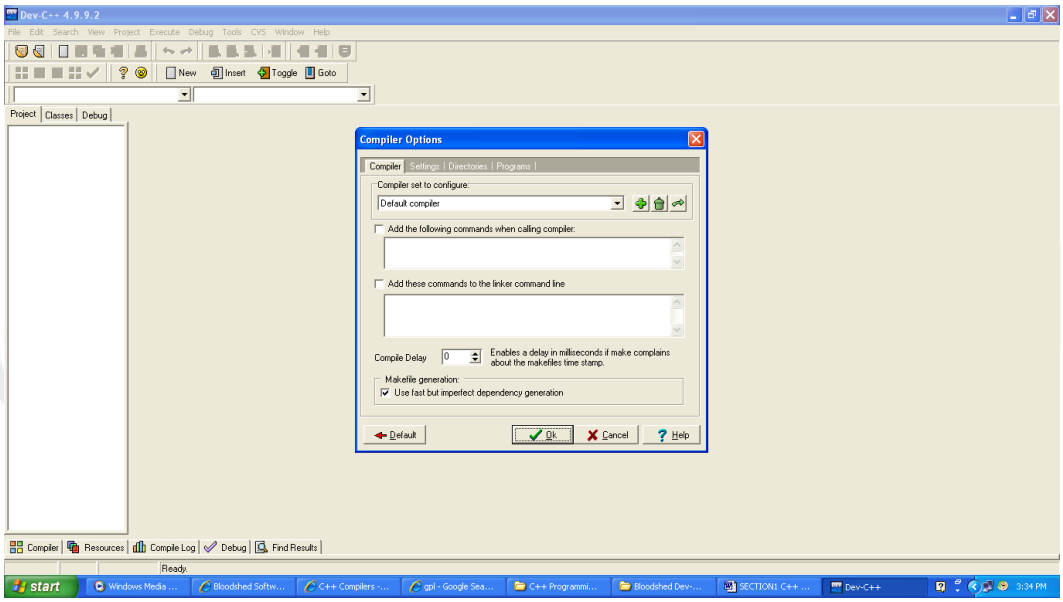


- In the "User's Default Directory" field, either type the path of your chice
- Leave all other defaults "as is".

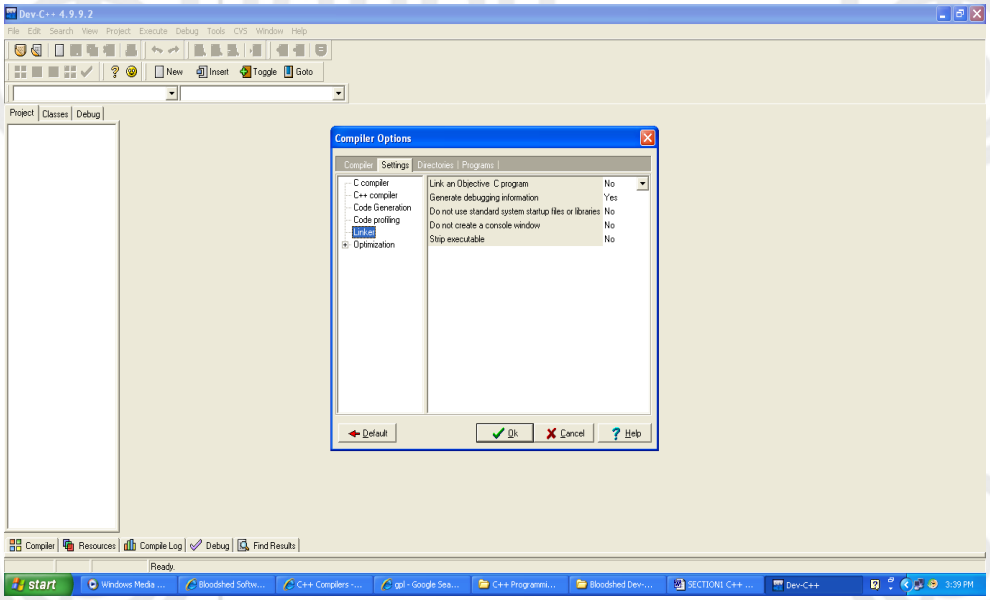
- Click the Ok button.

3. Before you start using Dev-C + + , you need to modify one of the default settings to enable you to use debugger with your programs:

   - Go to the "Tools" menu and select "Compiler Options".



- In the "Settings" tab, click on "Linker" in the left panel, and change "Generate debugging information" to "Yes":



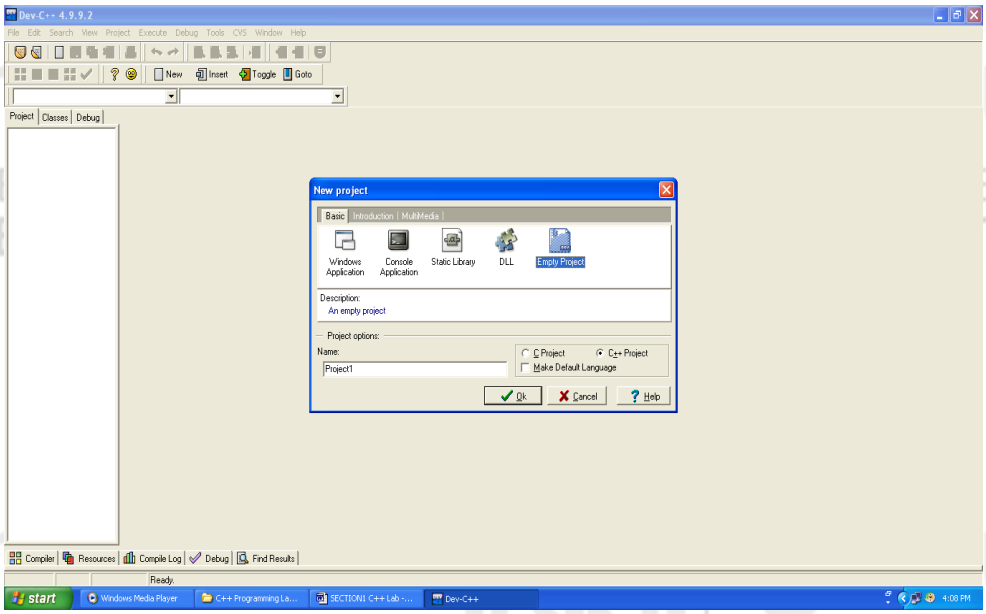
- Click "OK".

1.5 USING DEV C + + IDE

Create a new project/program in Dev-C + + :

Now you can use Dev-C + + for creating a “project/program” . A C + + project is like a container that is used to store all the elements that are required to compile a program.

- Go to the "File" menu and select "New", "Project".
- Choose "Empty Project" and make sure "C + + project" is selected.



Here you have to give your project a name. You should give your project a valid and meaningful filename. The name of your project will also be the name of your final executable.

A project may have more than one file in it. To add empty source files, follow one of two ways:

- Go to the "File" menu and select "New Source File" (or just press CTRL+N)

- Go to the "Project" menu and select "New File".

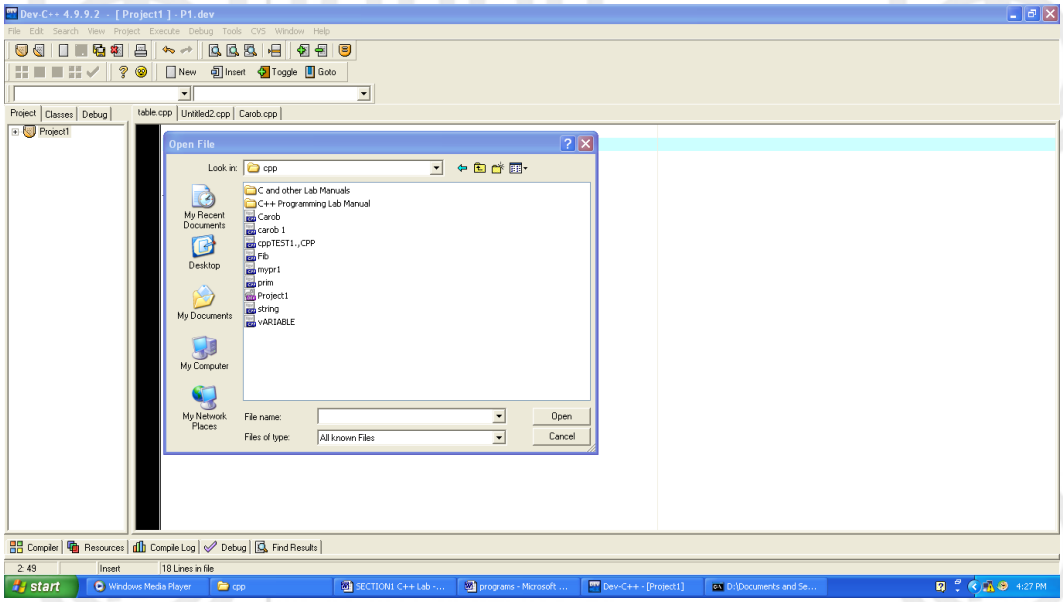
You should save your file befor compilation. Note that Dev-C + + will not ask for a filename for any new source file until you attempt to do any of the following actvity:

1. Compile
2. Save the project
3. Save the source file
4. Exit Dev-C + +

If you are having a pre-existing source files and want to add it to your project . You can do it with one of two following ways:

- Go to the "Project" menu and select "Add to Project"

- Right-click on the project name in the left-hand panel and select "Add to Project".



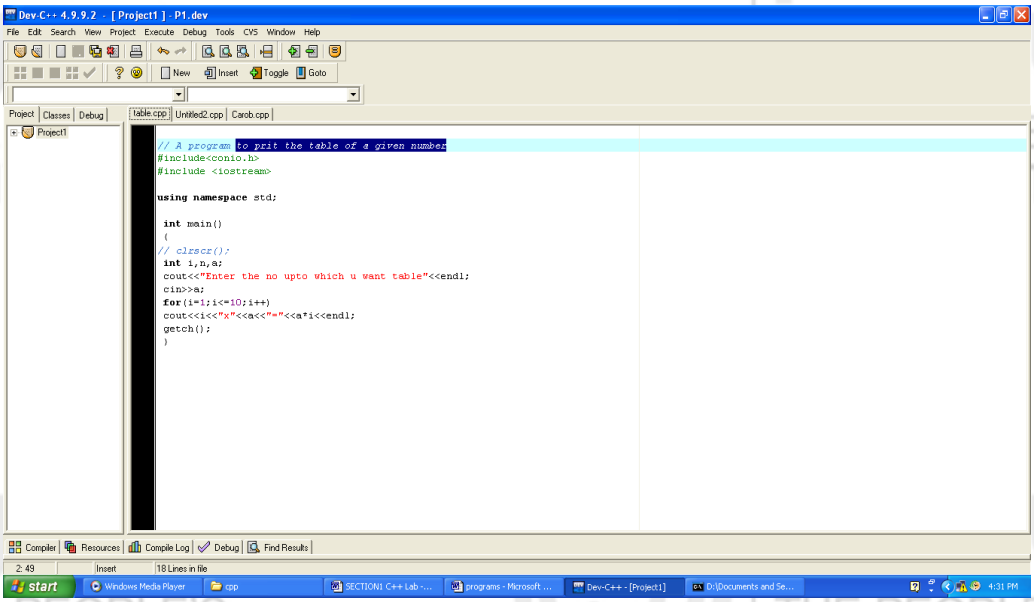
Now let us write our first program using Dev C + + . This program is to display message ***"Welcome to C + + programming".***

```
//Program to Print Hello message
 #include <iostream>
  using namespace std;
  int main(int n, char* Args[])
   {
       char name[20];
      cout << "Please enter your name:";
      cin>>name;
      cout << "Hello Mr./Miss. "<<name;
      cout<<"   Welcome to C++ programming"<<endl;
      system("PAUSE");
      return 0;

   }
```

The above program include the statement, **"using namespace std;"**. This line tells the compiler to use a group of functions that are part of the standard library (std). By including this line at the top of a file, you allow the program to use functions such as cout.

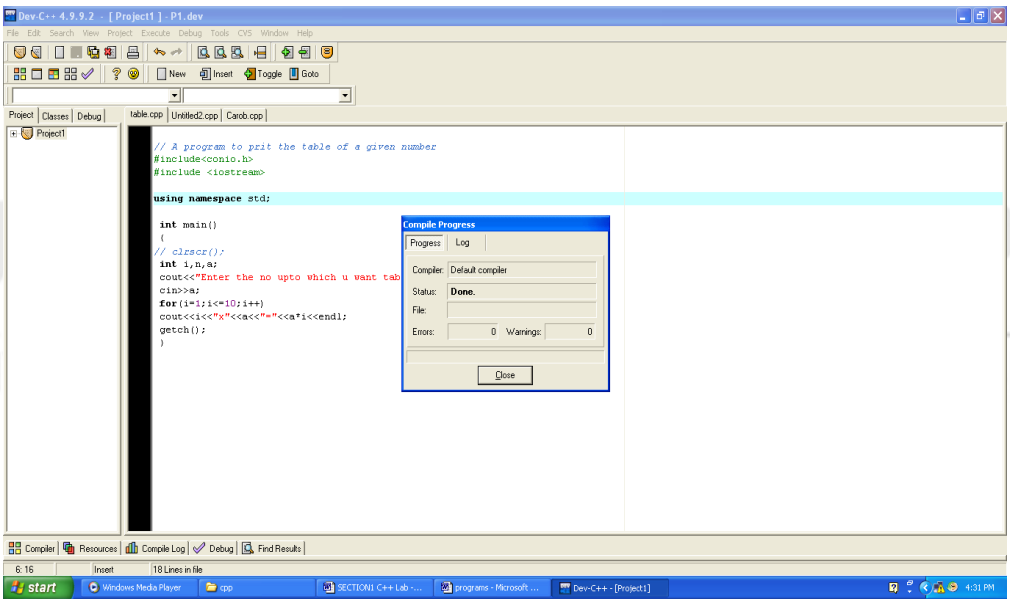
Let us try to execute above programe in Dev C + + . Type this program in Dev C + + IDE. Once you complete the writing, save the file with .cpp extention. Now you program is ready for compilation.



Compile program in Dev C + + :

Once your program coding is over, you are ready to compile.

- Go to the "Execute" menu and select "Compile" (or just press CTRL+F9).
- There is a chance that you will get some kind of compiler or linker error the first time you attempt to compile a project.
- Syntax errors will be displayed in the "Compiler" tab at the bottom of the screen.
- You can double-click on any error to take you to the place in the source code where it occurred.
- The "Linker" tab will flash if there are any linker errors. Linker errors are generally the result of syntax errors not allowing one of the files to compile.
- Once your project successfully compiles, the "Compile Progress" dialog box will have a status of "Done". At this point, you may click "Close".

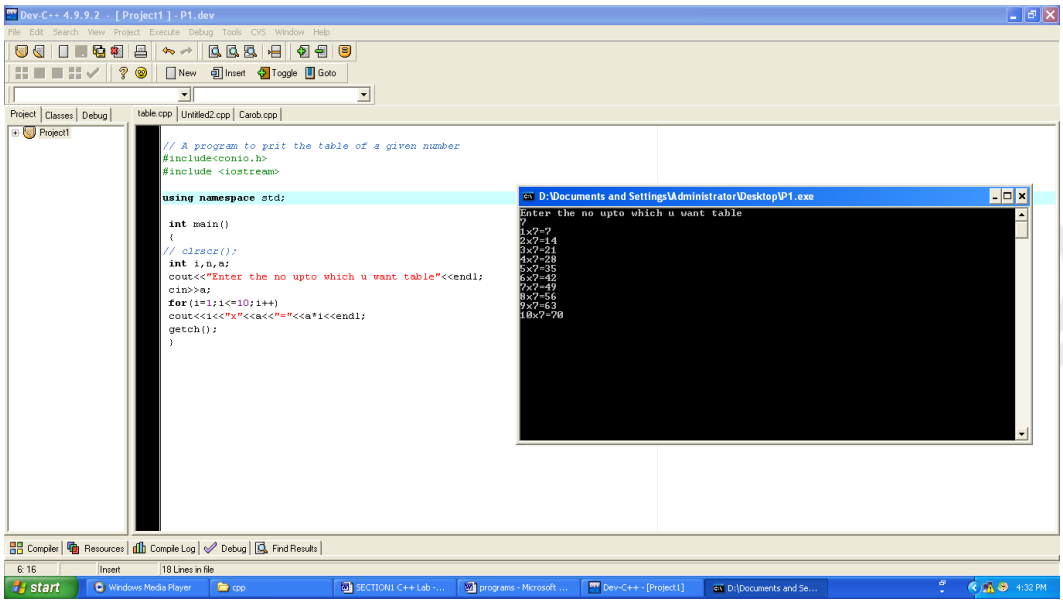


Executing Program in Dev C + + :

You can now run your program.

- Go to the "Execute" menu, choose "Run".

Note: If you have to pass command-line parameters to your program, go to the "Execute" menu, choose "Parameters" and type in any paramaters you wish to pass.



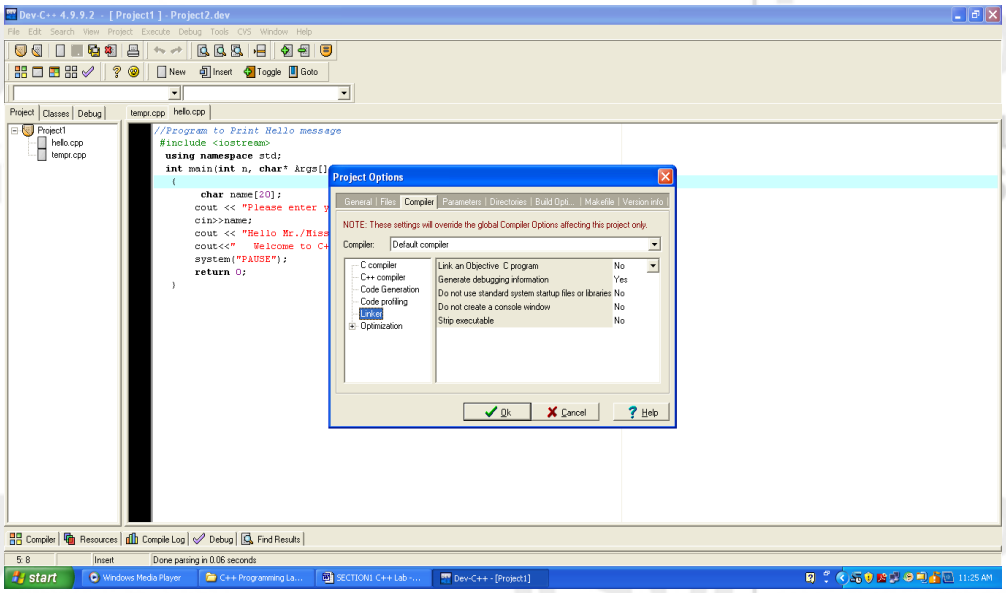
Removing Disappearing windows Problem:

If you execute your program, you may notice something peculiar; a console window will pop up, flash some text and disappear. The problem is that, if programs directly executed, console program windows close after the program exits.Not that in the above program system ("PAUSE"); command is used. If this command is not used, guess what will happen? The program would print the name entered to the screen and then immediately close without giving you the chance to read what was printed. But you need to see the result/output of the program. Here is the use of system ("PAUSE"); , it give you a chance to read what was actually printed. You may try the above program after removing/commenting, system ("PAUSE"); and can observe the advantage of this command.

Program Debug in Dev C + + .

For using debugging facility in Dev C + + , you have to first, make sure that you are using a project.Then follow the following steps:

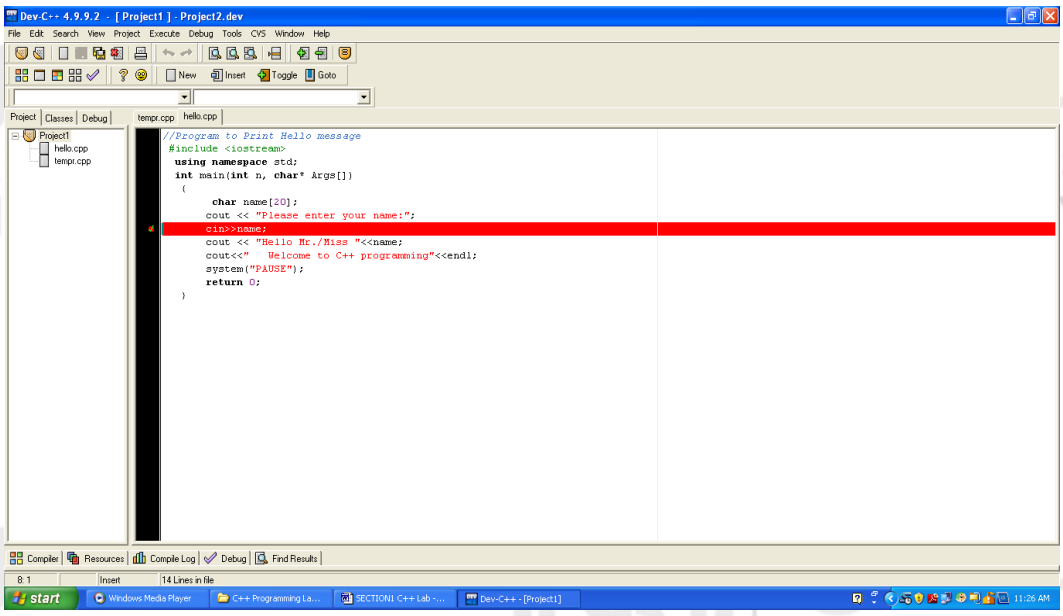
Goto Project Option> Compiler>Linker and set General debugging information to "yes".



After that, do a full rebuild (Ctrl-F11), then set breakpoint(s) where you want the debugger to stop (otherwise it will just run the program).

To set a breakpoint on a line, click on the gutter (the gray band on the left), or press Ctrl-F5.

Dev-C + + 's basic debugger functions are controlled via the "Debug" tab at the bottom of the screen. More advanced functions are available in the "Debug" menu of Dev C + + .



For using Dev C + + debugger functions you may use the following instructions:

- Click the "Run to cursor" icon to run your program and pause at the current source code cursor location.
- Click "Next Step" to step through the code.
- You can launch the debugger, by pressing F8 or clicking the debug button.
- Now you may start debugging. If exection of program goes well, the program will start executing step by step, and then stop at the first breakpoint.

- Then you may go through the code, entering function calls, by pressing Shift-F7 or the "step into" button, or stepping over the function calls, by pressing F7 or the "next step" button.
- You can press Ctrl-F7 or the "continue" button to continue execution till the next breakpoint.
- At any time, you can add or remove breakpoints.
- When the program stopped at a breakpoint and you are walking through the code, you can display the values of various variables in your program by putting your mouse over them, or you can display variables and expressions by pressing F4 or the "add watch" button and typing the expression or variable name.

A debugger basically runs a program while keeping track of its functions, variables and instructions. It is capable of stopping your program at a given moment, which is called breakpointing. It is necessary to set a breakpoint. So that the debugger may know where to stop. If you don't set a break point, the debugger runs the whole program and stop only after finishing of the program.

1.6 RUNNING EXAMPLE PROGRAMS USING DEV C + + IDE

In this section, we will practice some programming in C + + using Dev C + + . As you know, A C + + program is a collection of function, object, and type declarations. Every program must have a function int main() { ... } where the curly braces enclose a block, a sequence of declarations and statements ending in semicolons which are executed in order to solve the given problem.

A statement is an expression, block, or control statement which is used for decision making, looping etc. such as ***if, for, while, break, return***. Some types (std::string), objects (std::cout), and functions are defined in header files, requiring the line #include <*header*> before use. Items defined in the standard headers are in the namespace std.

In programming, before you try to receive input, you must have a place to store that input. The input and data are stored in variables. There are several different types of variables which store different kinds of information. When you tell the compiler you are declaring a variable, you must include the data type along with the name of the variable. The vname of variables should be meaningful. This will help you in tracing and debugging your program. Also if some body eles read your program, proper variable naming will them to understand your program.
Also using the right variable type is important for making your code readable and and efficient. You should use appropriate type when declaring variables, as some variables require more memory than others. For example to store numeric values, a float may be used to store an "inexact" value and inter may be used to store"exact" value.

Some time comilatiuon error create problem and irritation to programmer. Usually, times compilation errors may seem like a small, but as a beginner in C + + programming, you may not be familiar with the strictness of the format of the program (the syntax), hence you may tend to run into lots of complaints generated by the compiler. You avoid many of compilation error and add more clarity to your program by followinbg some basic principles given below:

- Just do not start learning C + + programming by compiling the examples. Try to understand the line by line code of examples given in your BCS-031 course or from some other reference locations.
- If there is something, you do not understand about a program, simply make some changes in the program and see the effect, in two to three attempts, you will come to know what actually that program is intended to do.
- With large and complex advanced programs that interact with the file system and uses advance features of OOP, you need to be much more careful about experimental program changes.
- Avoid jumping forward while program analysis. If the compiler indicates an error, or your program is not giving expect result, try to explore why this happened then try to fix the error. If a program compiles and runs without proper result, take the time to study each line and its purpose, so that problem can be traced.
- Do not write whole/many lines of code without compiling intermediate results of the program. Some time you will find that errors are unmanageable due to very large number of lines/code of the program.You may get confused to find the actual source of error. Hence, use compiling frequently to check your work.
- Upon reaching the end of main, the closing brace, you C + + program will return the value of 0 to the operating system. This return value is important as it can be used to tell the OS whether our program succeeded or not. A return value of 0 means success and is returned automatically.
- Declare a variable fefore its use, if you attempt to use a variable that you have not declared, your program will not be compiled or run, and you will receive an error message with information that you have made a mistake.
- C + + is case sensitive where use uppercases or lowercase letters matters. In C+ + , all language keywords, all functions and all variables are case sensitive. The word Age and age are two different things to the compiler. A difference in case between your variable declaration and the use of the variable in expressions/calculations will lead to undeclared variable error, while compilation.
- Give appropriate comments in program. Give comments at as many places as possible to explain structure and functionality of the program. This will also work to your advantage in ensuring more clarity and application.

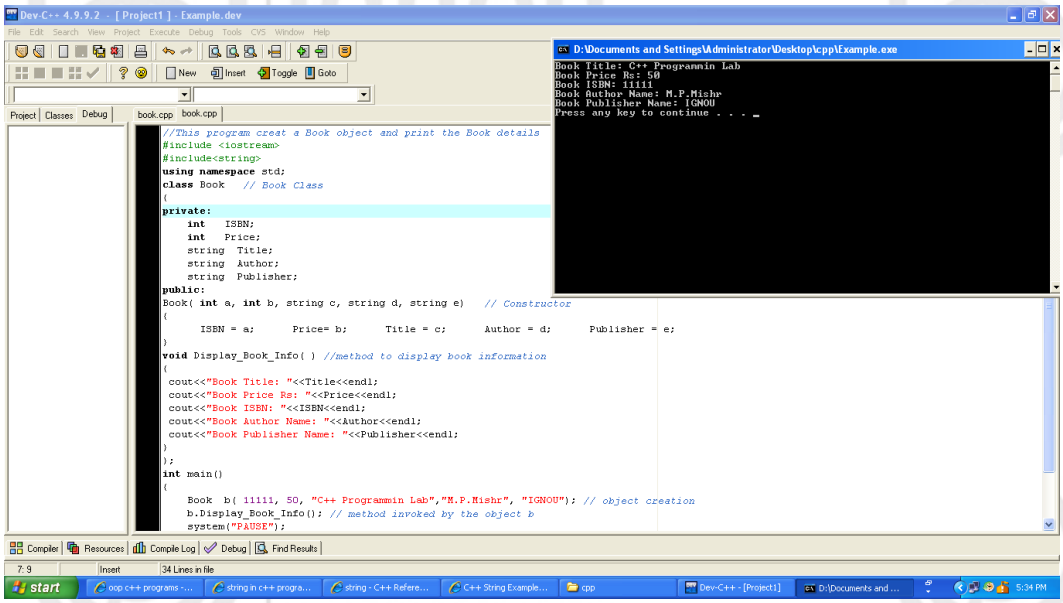
Now let us write our first program to crate a class in C + + .

Example1: This program l create a class named Book. This class is having following private data member:

- Title
- Author
- ISBN
- Publisher
- Price

Also the Book class is having following member functions to access/display the data members:

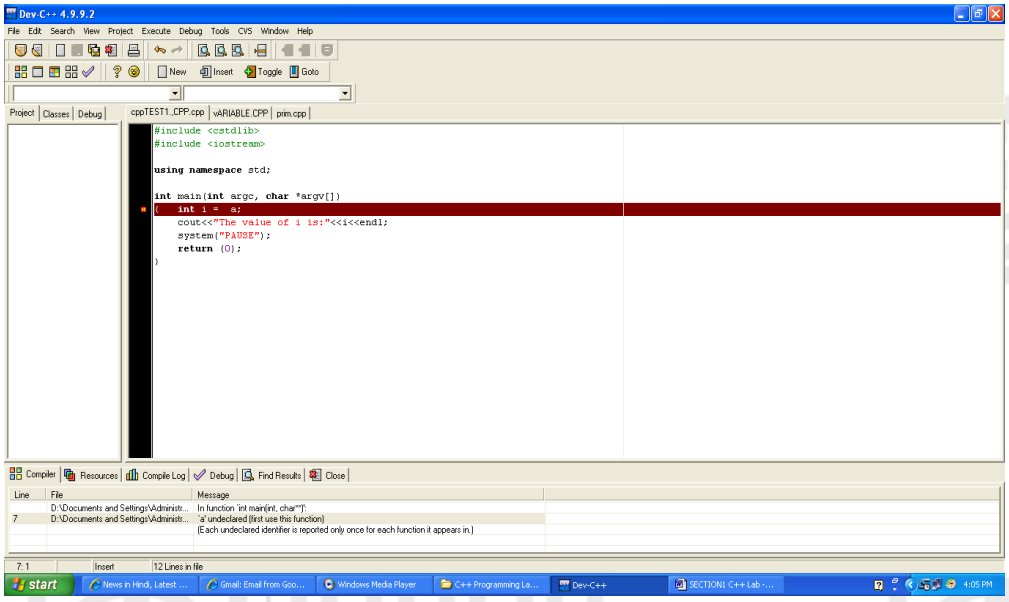
Display_Title() : It display the title of Book
Display_Price() : It display the price of Book
Display_Book_Info(): It display the title,price,author and publisher details of Book



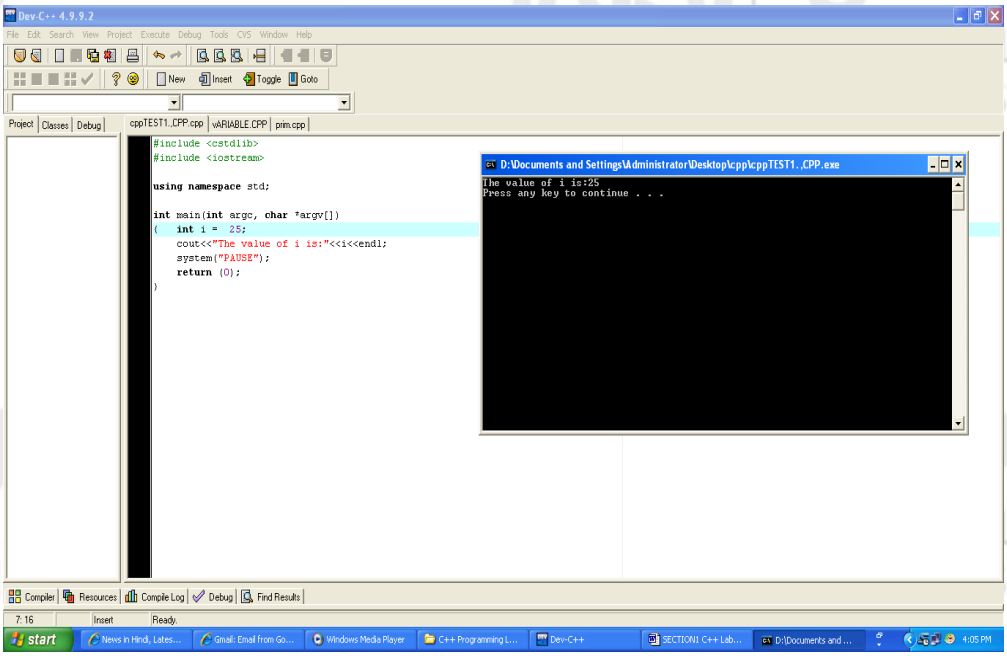
Example2: Try the following program code to run:

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{   int i =  'a';
    cout<<"The value of i is:"<<i<<endl;
    system("PAUSE");
    return (0);
}
```

You will get error message while compiling, because in this code an integer type variable **i** is assigned a character value which is a type mismatch.



Now try this program by assigning an integer value to i.



1.7 SESSION WISE PROBLEMS

We have allotted **ten** practical sessions for you to exercise. You have keep all program document in the file including all the steps involve in solving these problems. You must take output of pogram and should keep in your file. You are again advised to come prepared with your design of the following problems on paper in order to take maximum advantage of the Lab session. You are advised to make assumptions whereever necessary while solving these problems.

Session 1:

Problem1. Write a program in C + + to find and print the all basic data types of C + + .

Problem 2. Write a program in C + + to explain the use of for loop, while loop, swith-case, break and continue statements.

Problem 3. Write a program in C + + to find the maximu mark,average- marks and minimum marks obtained by a study in five papers as given below:

	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
Marks	50	70	65	80	56

Problem 4. Write a program in C + + that :

i) prints the factorial of a given number
ii) prints wheatehr a number is prime or not
iii) generate a Fibonacci series of 50 numbers.

Session 2:

Problem 5. Write a C + + program to create Student class with appropriate constructor and destructor. Also show what happen when you try to appent to access private data members from outside of the class.

Problem 6. Write a program in C + + to calculate the simple interest and compound interest for a given time period, given rate and for given principal amount using concept of class and member function.

Problem 7. Write a program in C + + to create an Employee class whichg read the following information from the keyboard:

Employee_id
Employee_name
Designation
Employee_Department
Years of experience

and display it on screen using display_info() method.

Session 3:

Problem 8. Write a program in C + + with class Rectangle with the data fields width, length, area and color .The length, width and area are of double type and color is of string type .The methods are set_ length () , set_width (), set_ color(), and find_ area (). Create two object of Rectangle and compare their area and color. If area and color both are same for the objects then display "Matching Rectangles" otherwise display "Non matching Rectangle".

Problem 9. Create a class Account with two overloaded constructors using C + + . First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, addresses, type of account and current balance. Account class is having methods Deposit (), Withdraw (), and Get_Balance().Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them. Also create appropriate destructors for the Account class.

Problem 10.Write a program in C + + to create a Queue class with insert () and delete () methods. Create two objects of Queue with 10 data item in both.Also define a method to find queue lenth.

Session 4:

Problem 11. Write a program in C + + to create a data base with the following items by using structure/class:

Name of the patient
Sex
Age
Ward_number
Bed number
Date of admission
List of Diease

Problem 12. Write a program in C + + to create a Player class. Inhermit classes Cricket _Player, Football _Player and Hockey_ Player from Player class. Create appropriate constructors and destructors for these classes.

Session 5:

Problem 13. Write a program in C + + to generate a database with the following details about the workers working in a manufacturing company, by using structure/class:

Name
DOB
Blood_Group
Height
Weight
Insurance_Policy_No
Office_address
DL_NO

Problem 14. Write a class Bookr and derive classes ReferenceBook and Magazines from it. Every book has a price, title, author and ISBN number. Implement conmstructors, destructos and appropriate methods to manage a Book Stor.

Problem 15. Write a C + + program to create Time class to impolement the working of a normal watch by creating approp constructor, destructor and member functiom.

Session 6:

Problem 16. Write a program which read two number ax and y then perform the following function

a. x + y
b. x - y
c. x * y
d. x / y

Design the function x +y and x – y in base class and design the function x * y and x / y in derived class

Problem 17. Write a program in C + + which create Student class with method to Display_Grade and manage Atrendance. Derive classes PG_Students and UG_Students from Student class.

Problem 18. Write a program by using inheritance to create the payroll of an organization by the following information:

Employee name
Employee Designation
Account Number
Date of Joining
Basic pay
DA, HRA, CCA
Deduction (PF, PPF etc)

Session 7:

Problem 19. Write a C + + program to create a class called DATE. Accept two

valid dates in the form dd/mm/yy. Implement the following operations by overloading the operators + and -. After every operation display the results by overloading the operator <<.
i. no_of_days=d1-d2; where d1 and d2 are DATE objects.d1>=d2 and no_of_days is an integer.
ii. d2=d1-no_of_days; where d1 is a DATE object and no_of_days is an integer.

Problem 20. Write a C + + program to create a class called COMPLEX and implement the following overloading functions ADD that return a COMPLEX number.
i. ADD(a, s2)- where a is an integer (real part) and s2 is a complex number.
ii. ADD(s1, s2)- where s1 and s2 are complex numbers.

Problem 21. Write a C + + program to create a class called MATRIX using a two-dimensional array of integers. Implement the following operations by overloading the operator== which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the operators + and – respectively. Display the results by overloading the operator<<.

```
if(m1==m2) {
m3=m1-m2;
m4=m1+m2;
}
else display error
```

Session 8:

Problem 22. Write a program which designs two classes and the calculate the division of first class private data with second class private data by using concept of friend function.

Problem 23. Write a program to calculate following function by using the concept of virtual function.

$$F(x) = 7x^5 + 9x^4 + 6x^3 + 2x^2 + 4$$

Session 9:

Problem 24. Write a C + + program to facilitate user to handle any chance of divide by zero exception.

Problem 24. Two vehicles are running on single track. If the vehicles are running in opposite direction there is a chance of collision. To avoid collisions write a C + + program using exception handling. You are free to make necessary assumptions.

Problem 25. Write a function template palindrome that takes a vector parameter and returns true or false according to whether the vector does not read the same forward as backward.

Problem 26. Creat a C + + Templet class for implementation of datastructure Queue functionalities.

Session 10:

Problem 27 . Implement the case study given in your course BCS-031 Block 3 Unit 4.

1.8 SUMMARY

This lab manual is for supporting the practice of C + + programming skill, based on BCS-031 course. This manual has detailed out how write and execute C + + programs. This lab manual started with discussion on need of a tool to write and execute C + + program. Using of the Dev C + + an IDE for C + + programming is explained with the help of explaining the steps to take in C + + program execution. Also how to debug C + + program in Dev C + + is explain in this manual. The best programming practice make a lot of impact in learing programming skill. This lab manual has hielighed how to write C + + programs to gain in terms of time saving and programming skill development. In the last section of this manual, a list of C + + programming exercised are given, which need to be solved learners by all of you for a complete understanding.

1.9 FURTHER READINGS

- Bjarne Stroustrup, *The C + + Programming Language*, 3rd edition,, Addison Wesley, 1997.
- John R Hubbard, *Programming with C + +* ,Schaum's Series, TMH publication, 3rd edition.Thinking in C + + , Sunil K. Pandey, S. K. Kataria
- Publication, Delhi, *Object-Oriented Programming in C + +* , 6th edition, Rajesh K. Shukla, Wiley India publication, 1st edition.
- www.indiastudychannel.com
- http://cs.fit.edu/~mmahoney/cse2050/how2cpp.html
- http://www.cprogramming.com/tutorial/lesson1.html
- http://en.wikipedia.org/wiki/List_of_compilers
- http://www2.research.att.com/~bs/compilers.html
- http://ceng230.ceng.metu.edu.tr/Compilers/dev-cpp.html