Turbo C++ 4.5 Tutorial|

By Sangeeth96|

Introduction

The first thing you should know before using this is that I'm not a professional programmer and is an amateur currently studying the basics. Why I uploaded is because my friends told me to do so. I basically got all the info from my classes and my friends told me to do this because they don't understand it clearly. So in here, I've tried my best to keep it in simple and pure words just for all the beginners. Hope you find it helpful. Comment if you have any doubts. Also check my facebook page for more.

System Requirements:

Windows 98/XP/Vista/7
Turbo C++ 4.5

Some of you will be wondering whether you need only these two main things for beginning. Guys, CPP isn't kind of a game or a big software and doesn't need additional processor, graphics or memory. If you plan to make huge software out of CPP, then you may think of getting a better PC.

About C++

The C++ programming language was developed by Bjarne Stroustrup at AT&T Bell Laboratories in USA. Through C++, Stroustrup sought to combine the best features of C and Simula67. C++ could be described as an extension of C with an additional class construct feature of Simula67. Since this class was a major addition to the original C language, Stroustrup initially called the new language 'C with Classes'. In 1893, the name was changed to C++. The ++ comes from the C increment operator ++ and suggests that C++ is an inherited version of C.

The Basics

Ok, So let's begin. Note that I'm using Turbo C++ 4.5 because it's the newsest one and is a GUI so it's really simple to use. Also, you can avoid getch like command. First, I'll start with a simple programme:

```
#include <iostream.h>
#include <conio.h>

int main()
{
      cout<<"My first cpp program";//To display the output
      return 0;
}
```

```
#include <iostream.h>
#include <conio.h>
```

You can copy the above program and paste it in the CPP window and run it to see the result.

So first, we can look at the #include stuff. The line beginning with a hash sign '#' is called pre-processor directive. To put it in simple words – It is a command which tells CPP to include key files into the CPP program. CPP along can't find all the commands by itself and it needs to refer to some place. So for that we use #include function. Here, we are referring to two keyfiles namely iostream.h and conio.h. Note that we use <> for including these files and they are important. So keep in mind that the less than and greater than brackets are used for this purpose in the basics only. Don't get confused with other brackets.

```
int main()
```

This line marks the beginning of the main function. The main function is the point from where all CPP programs start their execution.

{}

Then we start our program by a { bracket. Always place the program inside the {} brackets. After the program is over, don't forget to close it with }. These identify it as a program.

```
cout<<"My first cpp program";
```

As you can see, we get an output screen showing My first cpp program. So to display words or the values, we use cout. Always use cout<< for starting it and don't use cout>>. So if you want to display your own text or number, put it inside "".

```
//To display the output
```

Take a look at //. We can see that the text given in // doesn't appear in output screen. // is used for commenting a line. We use it to know what the line does and helps us understand each line in long programs.

If you are displaying to cout(s) like,

cout<<"Sangeeth96";
cout<<"e-galaxy";

If we print them, then it will look like sangeeth96e-galaxy. So If we want to bring one to the next line, we can use /n function. Give /n at the end of the first line (cout<<"sangeeth96/n";) or place it at the beginning of the succeeding line (cout<<"/ne-galaxy";)

Now at the end, you need to give return 0 always. After that put the } and thus close the program.

So, we can conclude that #include is used for inputting the key files, int for defining variable type, { } for starting and ending the program, // for commenting and cout for printing the text or variable.

Simple Sum

```
#include <iostream.h>
#include <conio.h>
int main()
{
      int a;
      a=2+3;
      cout<<"The sum is: "<<a;
      return 0;
}
```

So, copy this and paste it in CPP window and run it. You will get the output as The sum is 5. So what is this program for?

Here, we are doing the sum of pre-defined two numbers 2 and 3. We can't directly give the numbers if we want to define them. So here, we gave them as 2 and 3 so we can directly give them. Here, a is the sum of the numbers and it's a variable. To input a variable which is integer type, type int a or int n1,n2,n3…… or anything like even intarea,radius……..and so on. Here we only need one variable i.e, for sum only. So let us take sum be a. So we can type int a;

Since we are already defining numbers in the program, we only need to give direct calculations i.e, a=2+3. Now we need to display the result. So we can give cout<<c. But displaying the result only confuses the viewer so let us add a custom message to the cout function i.e, cout<<"The sum is: "<<a;. After this, give return 0 and close the program.

NOTES:

- For printing a variable, we need not give "". We can give as much as cout we want but make sure you put << after each one when separating a text and a variable.
- Since we are just printing the results of a pre-defined calculation, we can add a message at first letting the viewer understand we are performing a specific calculation.
- Remember that CPP inputs variables you give case-sensitively. So, if you input a variable 'a' in the beginning and while performing a calculation or printing the variable, you must always give 'a' according to whether it is uppercase or lowercase. If you give 'A', then it will result in errors.
- Note to give variables to the **left** side only. CPP always gives values to variables from right to left. So, if you type 2+3=a instead of a=2+3, it will result in errors. So, be sure to check whether all the variables are given in the Left Hand Side.

Practice Corner

So far, we've learned how to display a message and to find sum of pre-defined numbers. Now try out the following in C++;

1. Display your name and school.
2. Find product of 10 and 5

Cin>>

```
#include <iostream.h>
#include <conio.h>
int main()
{
      int a,b,c;
      cout<<"Enter two numbers: ";
      cin>>a>>b;
      c=a+b;
      cout<<"The sum is: "<<c;
      return 0;
}
```

The above program is used to find the sum of any two numbers given by the user. Try copying it to CPP dashboard and run it.

```
cin>>a>>b;
```

Here, we are using a special command, the cin>> for the program. First we enter three variables namely a,b,c (you can name the whatever you want). “a” is the first number and “b” is the second number. “c” is the output or result.

So, we give a message asking the user to enter any two numbers which are integers. After giving the cout message, we type a new command, the cin>> function. It’s method is like this “cin>>variable;”. We give a and b as the variables because they are the numbers’ whose sum we want to find. Note that we use >> for cin function and not <<. <<is used with cout and >>is used with cin and so don’t get confused. Here, we give cin>>a>>b;

So, if we are running the program, it will display the message Enter two numbers. Then the user needs to give the numbers. First type a number and press Enter key. Now type the second number and again hit the enter key.

We need to find the sum, so we give c=a+b where c is the sum of a and b. Now, give a cout command to show a message like ‘the sum is’ and then print ‘c’. Now close the program.

NOTES

- Only give operations like c=a+b only after the cin function. CPP doesn’t automatically arrange the program so you must give all the lines step-wise.
- The Cin command always have a double greater than “>>” symbol. Cout uses double less than “<<” symbol. So, don’t get confused with cout and cin. Always use the symbols which are meant for each command.

Practice Corner

Now try out these problems.

1. Find the product of any three numbers. (Hint: Give 4 variables, 3 for the numbers and one for the product)
2. Create a programme to find the circumference of a circle of a custom radius. (Hint: instead of inta,b,c; give float a,b,c. E.G. float radius,circumference.

Operators

You can learn more about operators and its various types from your textbook. Operators are used for various types of operation with variables and constants. Some of the operators used in C++ are assignment operator, arithmetic operators, compound assignment operators, increase and decrease operators, relational operators, logical operators, conditional operators, comma operator, bitwise operator and sizeof() operator. We can just cruise around some of the operators, then we can learn and practice some C++ programs and learn the if and else statements. Here, we will look onto only some of the operators. If you want to know all, refer to your textbook.

Assignment Operator

'=' is known as the assignment operator. We use = in maths to show equality between two numbers, variables or expressions. But in C++, = is used only for assigning a value to a variable, constant or a mixture of these. For example,

x = 2

Here, x is given the value 2. So, if you want to show equality, you cannot use =. There's another operator you can use and we will come to it later.

Arithmetic Operators

The five arithmetic operators used in C++ are +(for addition), -(for subtraction), *(multiplication), /(division) and % or modulus. % is used for finding the reminder of the division. For example, instead of 5/2, if we give 5%2, we will get the output as 1 because the reminder is 1.

Relational Operators

Relational operators are used to compare two expressions. The result of a relational operation is a Boolean value that can only be true or false. The relational operators used in C++ are given below. The operators are == (equal to). != (not equal to), > (greater than), < (less than), >= (greater than or equal to) and <= (less than or equal to). == is used to find the equality of two values. For example, for the expression x==5, if we give 5 as the value for x, we will get true value as 5 = 5. We cannot give x=5 for checking in C++ as it will assign the value 5 to x, not check the similarity.

.....This section will get updated, you can continue on to the if else statement page and try out the programs which are related to the given operators.

if and else statement

Now, we can learn two new commands in CPP – if and else. In this tutorial, we'll learn to create a simple program which can find the greater of any two numbers. Copy the code below and paste it in the CPP window. Run the program.

```
#include <iostream.h>
int main()
{
      int num1,num2;
      cout<<"Enter two nos: ";
      cin>>num1>>num2;
      if (num1>num2)
      {
            cout<<num1<<" is greater";
      }
      else
      {
            cout<<num2<<" is greater";
      }
      return 0;
}
```

As you have seen, this program is used to identify which number among any two numbers that the user specifies is greater. Here, we created this program with the help of if and else statements. The general format is:

```
      if (condition)
        Statement1;
      else
        statement2;
```

The if statement is written as given above. After if, inside two curved brackets, you must give the condition, e.g:- here, it is num1>num2. Then, you must give a message if the condition is true again inside two curly brackets {}. It can be a cout message like The number 1 is greater. Here, we gave order to C++ to check whether the first number or num1 is greater than the second number or num2. After checking, if num1 is greater than num2, it will display the message given after the first curly braces. If it's not, you must give another message to show that the second number is greater. Since there are only two numbers in this program, we need not add else if and then the condition. If the first number is not the greater of the two, then it is definitely the second number which is larger. So, you can give else. 'else' doesn't need any condition since it is directly displaying the left value. So, you can give a message like 'the second number is greater'

inside the curly braces. Be sure to close the braces after you have added your message. If there are more than two numbers, you can use the else if statement and define the condition. At last, you can use the else statement.

NOTES

- When using the if statement, there is no need to use semi colon (;).
- Also, put the condition inside two curved brackets.
- Remember to put a brace inside the if statement to show additional statements and remember to close it. More braces can confuse you, so do accurately.

Practice Corner

We learned how to use the if else statement. Now, why don't you try out some of these easy problems to get you on with?

1. Check whether a number is positive or negative (Hint: less than 0 gives negative)
2. To display grades for a student based on his/her marks totally. (Hint: <40 – failed; >=60-C; >=75-B; >=90-A; >=98-A+)

Ask Sangeeth96 to know more.
drakeztar@yahoo.com.....

Keep connecting through my twitter and facebook accounts…

Guide Incomplete…