**A Simple C Program:**

#include<stdio.h>

#include<conio.h>

int main()

{

printf(“Welcome to C!!!”);

getch();

return 0;

}

**Output:**

Welcome to C!!!

**#include<stdio.h>**

* Is a directive to the C preprocessor. Lines beginning with **#** are processed by the preprocessor before the program is compiled. This specific line tells the preprocessor to include the contents of the standard input/output header**<stdio.h>** in the program.This header contains information by the compiling calls to standard input/output library functions such as **printf**.

**int main()**

* Is a part of every C program. The parentheses after main indicate that main is a program building block called a function. C programs contain one or more functions, one of which must be main.

Every program in C begins executing at the function main.

The left brace { must begin the body of every function. A corresponding right brace must end each function. This pair of braces and the portion of the program between the brace is called **BLOCK.**

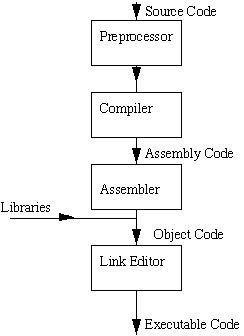
**printf(“Welcome to C!\n”);**

* Instructs the computer t perform an action, namely to print on the screen the string of characters marked by the quotation marks.

**return 0;**

* Is included at the end of every main function. The keyword return is one of several means we will use to exit a function. When the return statement is used at the end of main as shown here, the value 0 indicates that the program has terminated successfully.

**THE C COMPILATION MODEL**



**Some Common Escape Sequence**

|  |  |
| --- | --- |
| Escape Sequence | Description |
| \n | Newline. Position the cursor at the beginning of the next line. |
| \t | Horizontal Tab. Move the cursor to the next tab stop. |
| \a | Alert. Sound the system bell. |
| \\ | Backslash. Insert a backslash character in a string |
| \” | Double quote. Insert a double quote character in a string |

**What are variables?**

Variables in C are memory locations that are given names and can be assigned values. We use variables to store data in memory for later use. There are 2 basic kinds of variables in C which are numeric and character.

* **Numeric variables**

can either be integer values or they can be Real values. Integer values are whole numbers without a fraction part or decimal point in them. Real numbers can have a decimal point in them.

* **Character variables**

are letters of the alphabet as well as all characters on the ASCII chart and even the numbers 0 - 9. Characters must always be put between single quotes. A number put between single quotes is not the same thing as a number without them.

**What are constants?**

The difference between variables and constants is that variables can change their value at any time but constants can never change their value. Constants can be useful for items such as Pi or the charge on an electron. Using constants can stop you from changing the value of an item by mistake.

**Declaring variables**

To declare a variable we first put the type of variable and then give the variable a name. The following is a table of the names of the types of variables as well as their ranges:

**Some Data types:**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Format | Meaning | Description |
| int | %d | Integer.Whole number | (3,7,+8.-9,100000) |
| Char | %s | String.A string in is considered as a Series of characters | “Ms.”,”HI”,”Love”,”143” |
| Char | %c | Character.Single letter, symbol | ‘b’, ‘2’,’\*’ |
| Float | %f | Float.A number with decimal point | 5.5,9.0,-5.1212,+8.56 |
| Double | %f | Double.For bigger or larger numbers with decimal point | 12314.1444,7864534.23 |

**Adding Two Integers:**

#include<stdio.h>

#include<conio.h>

int main()

{

int Int1,Int2,sum;

printf(“Enter first number:”);

scanf(“%d”,&Int1);

printf(“Enter second number:”);

scanf(“%d”,&Int2);

sum=int1+int2;

printf(“Sum is %d”,sum);

getch();

return 0;

}

**Output:**

Enter first number: 5

Enter second number: 5

Sum is 10

**Operator Precedence:**

|  |  |  |
| --- | --- | --- |
| Operator | Operation(s) | Order of Evaluation(precedence) |
| \* | Multiplication | Evaluated first. If there are several, they are evaluated left to right |
| / | Division |
| % | Remainder |
| + | Addition | Evaluated next. If there are several, they are evaluated left to right. |
| - | Subtraction |

**Equality and Relational Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| Standard algebraic equality operator or relational operator | C equality or relational operator | Example of C Condition | Meaning of C Condition |
| **Equality Operators** | | | |
| = | == | X==y | X is equal to y |
| ≠ | != | X !=y | X is not equal to y |
| **Relational Operators** | | | |
| > | > | X >y | X is greater than y |
| < | < | X < y | X is less than y |
| >0 | >= | X>=y | X is greater than or equal to y |
| < | <= | X<=y | X is less than or equal to y |

**Exercises:**

/\*Program that compute for the area of circle and display the result.Formula:A=πr2 where Pi(π) is approximately equal to 3.1416.\*/

#include<stdio.h>

#include<conio.h>

#define pi 3.1416

#define p printf

#define s scanf

int main()

{

int r;

float c;

p(“Enter radius:”);

s(“%d”,&r);

c=pi\*r\*r;

p(“\nCircumference is %.2f”,c);

getch();

}

Enter radius:4

Circumference:50.27

/\*Program that converts input Fahrenheit degree into its Celsius degree equivalent. Use the formula;C=(5/9)\*F-32.Display the result.

Enter Fahrenheit:

The Celsius is:

#include<stdio.h>

#include<conio.h>

#define p printf

#define s scanf

int main()

{

float c,f;

p(“Enter fahrenheit:”);

s(“%f”,&f);

c=(f-32)\*5/9;

p(“\nThe Celsius is %.2f”,c);

getch();

}

Here are the relational operators, as they are known, along with examples:

> greater than 5 > 4 is TRUE

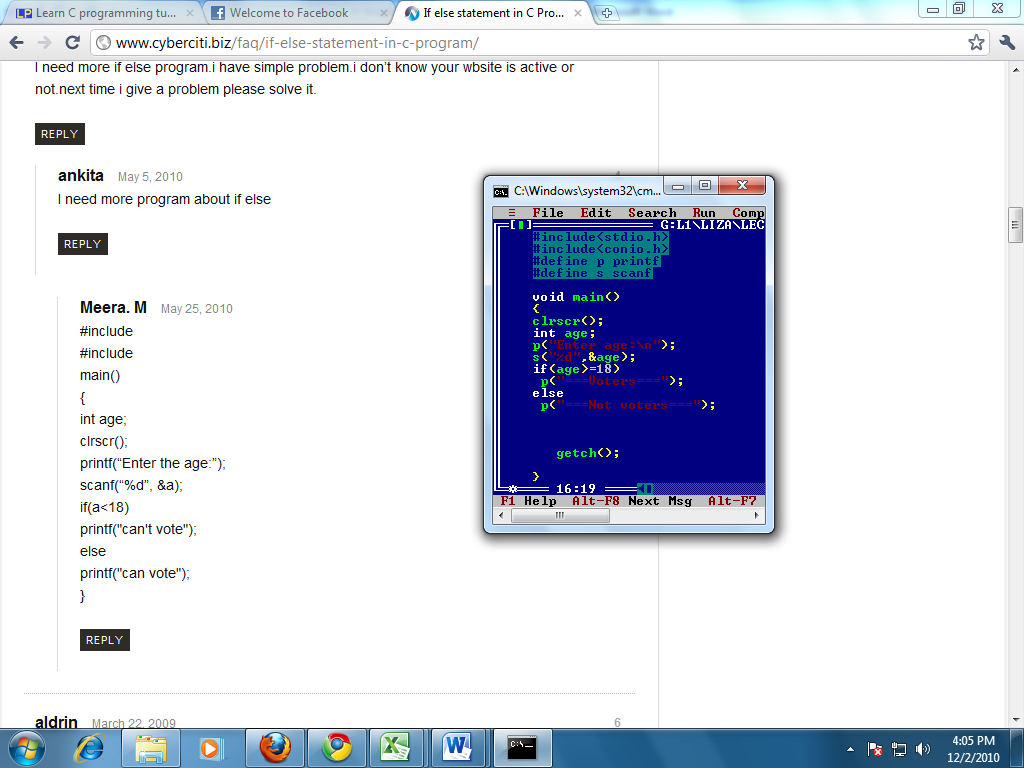
< less than 4 < 5 is TRUE

>= greater than or equal 4 >= 4 is TRUE

<= less than or equal 3 <= 4 is TRUE

== equal to 5 == 5 is TRUE

!= not equal to 5 != 4 is TRUE

**The if statement**

**Output:**

Enter age:

20

===Voters===