**Learn C by example in just 5 hours.C tutorial on-line.**

Have you always wanted to master a programming language. Well today if you are glancing at this page you have chosen a language which perhaps without doubt is the most versatile. But to learn C for say basic programmers is a challenge. While the old basic used interpreters C uses compilers and basically is very portable. But let quit all this jibrish and get to the heart of this page. I say you can learn C programming in 3 hours. Well atleast the basics that will help you to build more powerful programs.You say I can't show you C in 5 hours. Well let's test that ...

**A simple hello program.(demonstrates the const function in all c programs--the main() function.)  
(example-1)  
main()  
{  
puts("hello world guess who is writing a c program");  
return(0);  
}**

**That's it. In all c programs there is a main function which is followed by a { and closed by a } after a return()function.It doesn't have to be return(0) but that depends upon the type of c compiler you have. Check your compiler before you start your programming.**

**You saw above that puts function is used to put a whole sentence on the screen; but are there functions that will put characters on the screen/take characters: Yes and next is a table of what they are and what they do. Read them and the examples that follow.**

|  |  |
| --- | --- |
| **getchar()** | **Gets a single character from the input/keyboard.** |
|  |  |

|  |  |
| --- | --- |
| **putchar()** | **Puts a single character on the screen.** |
|  |  |

**The printf function is a function used to print the output to the screen.printf() needs to know if the output is an integer,real,etc example-2  
main()  
{  
printf(hello);  
}  
Assuming hello was defined earlier say by #define hello "Hello!" the output is Hello!. But if the output is an integer then %d has to be attatched to the printf statement.**

**This above can be shown as printf("I am %d years old",12) which will result in the following result:I am 12 years old**

**The %d tells that an integer is to be placed here.**

**Now we will look into a function called scanf().This lets you input from the kewyboard and for that input to be taken by the program and processed.Once again it is important to tell scanf() what type of data is being scanned.**

**Here is an example of a program that demonstrates both scanf and printf in unison.  
example-3**

**main() {  
int count;  
puts("Please enter a number: ");  
scanf("%d", &count);  
printf("The number is %d",count);  
}**

**That concludes the first hour of your tutorial.Now this is a list of data type identifiers.**

**%f=float %c=char %s =s tring %e=inputs number in scientific notation.**

**As you saw in the first hour of our tutorial c is a language in which you program using functions. Functions are usually identified by the following characteristic:>> functionname() In c the main() function is essential. Think of it as a constant function for all your programs and all other functions can be accessed from the main().Before I show you how we do that let us have an example where we want to pause a program before the screen is changed. This would involve the foll- owing procedure:>> write a main function then use puts function to put statements on the screen like we did in section 1 above and then before the next set of puts statements declare a pause.**

**This is how it is done:  
  
example-4   
main()  
{  
puts("hello there");  
puts("what is your name?")  
pause()  
puts("It is nice to meet you")  
}  
pause();  
{  
int move\_on;  
printf("press entere to continue");  
move\_on=getchar();  
return(0);  
}  
  
This above will pause until a key is pressed on the keyboard. Granted that the above program makes no sense from a practical point of view but I want to show is the use of another function inside the main function.**

**C has many functions that comes with it. See your compiler manual to see what you have.Now we are going to look at conditions in c programming:>> the if command and do command.**

**Here is an example of th if command:  
  
example-5   
main()  
{  
float cost,tax,luxury,total;  
luxury=0.0;  
printf("Enter the cost of the item: ");  
scanf("%f", &cost);  
tax=cost\*0.06;  
if(cost>40000.0)  
luxury=cost\*0.005;  
total=cost+tax+luxury;  
printf("the total cost is %0.2f",total);  
}  
  
This is a simple example of one if statement. Another If statement is the if -else statement. This can be shown as this   
  
example-6   
if(cost >40000)  
{  
luxury=cost\*0.005;  
printf("The luxury tax is %.2f",luxury);  
}  
else  
{  
puts("There is no luxury tax for the items");  
luxury=0.0;  
}  
  
Now the format a do statement is as follows:  
  
do  
{  
instruction;  
instruction  
}  
while(condition);  
  
The format for a FOR statement is as follows:  
  
for(initial=value;condition;increment)  
instruction;  
  
Now for an example:  
  
example-7   
main()  
{  
int row,column;  
puts("\t\tMY Handy multipication table");  
for(row=1;tow<=10;row++)  
{  
for(column=1;column<=10;column++)  
printf("%6d", row\*column);  
putchar('\n');  
}  
}  
  
The output is a multipication table of 10x10 size.  
  
example-8   
main()  
{  
int temp;  
float celsius;  
char repeat;  
do  
{  
printf("Input a temperature:");  
scanf("%d", &temp);  
celsius=(5.0/9.0)\*(temp-32);  
printf(%d degrees F is %6.2f degrees celsius\n",temp, celsius);  
printf(("do you have another temperature?");  
repeat=getchar();  
putchar('\n');  
}  
while(repeat=='y'|| repeat=='y');  
}  
  
This shows you to how to use the do command for conditional programming in c.**

**Now we are in our 3rd hour.**

**Now we will concentrate on arrays:  
  
What is a flag?**

**A flag is an algorithm that informs the program that a certain condition has occured.  
  
example-9**

**main()  
{  
int temp;  
float celsius;  
char repeat;  
char flag;  
do  
{  
flag='n";  
do  
{  
if(flag=='n')  
printf("Input a valid temperature :");  
else  
printf("input a valid temperature,stupid:");  
scanf("%d",&temp);  
flag='y';  
}  
while (temp<0||temp >100);  
celsius=(5.0/9.0)\*(temp-32);  
printf("%d degrees F is %6.2f degrees celsius\n",temp,celsius);  
printf("Do you have another temperature?");  
repeat=getchar();  
putchar('\n');  
}  
while (repeat=='y' || repeat=='Y");  
}  
  
That was an example of how flags work.**

**What is the break command?**

**The break command ends the loop in which it is placed just as if the while condition, or the condition in a for loop becomes false.**

**How to declare an array?**

**An array can be defined as follows:  
  
int temp[5]={45,56,12,98,12};  
  
This would mean the following:  
  
temp[0]=45....temp[4]=12  
  
This was a single dimension array with 5 elements of the integer type.If you wanted to depict float variables just use float temp instead of int temp.**

**Let us now see an example of using an array for two tasks.**

**main()  
{  
int temps[31];  
int index,total;  
float average,celsius;  
total=0.0;  
for(index=0;index<31;index++)  
{  
printf("enter temperature #%d:",index);  
scanf("%d",&temps[index]);  
}  
for(index=0;index<31;index++)  
total+=temps[index];  
average=total/31.0  
printf("average is:%f\n\n", average);  
puts9"fahrenheit\tcelsius\n");  
for(index=0;index<31;index++)  
{  
celsius=(5.0/9.0)\*(temps[index]-32);  
printf("%d\t\t%6.2f\n",temps[index],celsius);  
}  
}**

**Now I am going to show you how to pass an array. When you pass an array you are actually passing the address of the array.  
  
example-10  
  
#define count 31  
main()  
{  
int temps[count];  
int index;  
float celsius;  
for(index=0; index< count;index++)  
{  
celsius=(5.0/9.0)\*(heat[index]-32);  
printf("%d\t\t%6.2f\n",heat[index],celsius);  
}  
}**

**Now we are in the fourth hour of our tutorial.We are now going to look at 1)comparing strings 2)determining string lengths. 3) combining strings 4)structures.**

**Comparing 2 strings:>> In c it is not possible to directly compare two strings so a statement like if (string1==string2) is not valid.**

**Most c libraries contain a function called the strcmp().This is used to compare two strings in the following manner.  
  
if(strcmp(name1,name2)==0)  
puts("The names are the same");  
else  
puts("The names are not the same.");**

**Determining string length.:>> This is done using the strlen() function.**

**a simple programming bit showing this function looks like this:**

**gets(name);  
count=strlen(name);  
printf("the string %s has %d characters",name,count);**

**Combining strings:>>We use the function strcpy() an example follows:  
  
Example-11**

**strcpy(name,"Adam");  
strcpy(name1,"and eve");  
strcat(name,name1);  
puts(name);**

**The assumption being that adam and eve are two values of the variables name1 and name2. The end result is the combination of the 2 names.**

**What are structures?**

**A structure variable is a collection of other variables comprising different types.**

**What are pointers?**

**Ponters are variables which refer to the memory locations of other variables.  
  
This is how a structure is defined.  
  
example-12**

**struct cd  
{  
char name[20];  
char description[40];  
char category[12];  
float cost;  
int number;  
};  
main()**

**Notice how the main function comes after the definition of the structure. In the example above the cd was a cd disk and I was writing the definition of a cd collection program.**

**Now in the fifth hour I will show you how to output your data onto a disk.After all what is the use of the program if you can't save output to a disk.**

**Inorder to do this we have to use a pointer. The pointer in this case is FILE. The syntax to declare a file is :FILE\*file\_ponter;**

**The link between your program, the file and the computer is established with the fopen() function using the syntax shown below:   
pointer=fopen("FILENAME","mode");**

**For example to create a file by the name cd.dat we do the following:  
  
FILE\*cdfile;  
cdfile=fopen("CD>DAT","w");  
If you will be reading from the file above use "r" instead "w" in the   
second sentence.**

**In order to rpint information use the following command:  
FILE\*cdfile;  
cdfile=fopen("PRN","w");  
A file is closed by using the fclose() command.Next we will look at an exam ple of reading from a file.  
  
example-13**

**#include "stdio.h"  
main()  
{  
FILE\*fp;  
int letter;  
if((fp=fopen("MYFILE","r"))==NULL)  
{  
puts("Cannot oepn the file");  
exit();  
}  
while((letter=fgetc(fp)) !=eof)  
printf("%c",letter);  
fclose(fp);  
}  
  
The eof statement means end of file and this is included in the stdio.h header file which was declared at the start of the example. The stdio.h header file is one of many that comes with your compiler. So check your compiler specifics for other header files which will help perform other functions.**

**Now that you went through this tutorial you should be in a position to write simple programs and save it to a disk so you can give it your friends or even your boss. In no way the depth of c can be done in 5 hours but the nut and bolts can be learned that fast.Wher e you go from there depends upon your ambitions and hard work.  
  
This page is maintained by Francis Thottungal.**