Learn 'C' by Example

A simple hello program (demonstrates the constant function in all c programs--the main() function.)
(example-1)

```
#include<stdio.h>
main()
{
   puts("hello world guess who is writing a c program");
 }
```

That's it. In all 'c' programs there is a main function which is followed by a { and closed by a }

You saw above puts function is used to put a whole sentence on the screen; but there are functions that will put characters on the screen/take characters: Yes and next are few more functions below.

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
(example-2)

```
#include<stdio.h>
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 to give input from the keyboard 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)

```
#include<stdio.h>
main()
{
   int count;
   puts("Please enter a number: ");
   scanf("%d", &count);
   printf("The number is %d",count);
}
```

Let's see the list of data type identifiers.

%f=float, %c=char, %s =string, %e=inputs number in scientific notation.

As you saw above tutorial, 'c' is a language in which you program using functions. Functions are usually identified by the following characteristic :>> function name () 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 you learn 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 following 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)

```
#include<stdio.h>
main()
{
  puts("hello there");
  puts("what is your name?");
  pause();
  puts("It is nice to meet you");
}
```

```
pause()
{
 int move_on;
 printf("press enter to continue");
 move_on=getchar();
 }
```

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 just want to show this is the use of another function inside the main function.

C has many functions that come with it. Now we are going to look at conditions in 'c' programming:>> the if statement.

Syntax of *if* statement

```
If (condition)
  {
     Statement1;
     Statement2;
  }
```

Here is an example of the *if* statement:
(example-5)

```
#include<stdio.h>
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)                //brace are not necessary for single statement
  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 if -else statement. This can be shown as this

Syntax of *if else* statement

```
If (condition)
  {
    Statement;
  }
Else
  {
     Statement;
  }
```

Example of *if else* statement
(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;
 }
```

The format for a *for* statement is as follows

```
for (initial=value; condition; increment)
   {
      Statement;
   }
```

Now for an example:
(example-7)

```
#include<stdio.h>
main()
{
   int row,column;
   puts("\t\tMY Handy multipication table");
```


```
 for(row=1;row<=10;row++)
  {
    for(column=1;column<=10;column++)
    printf("%6d", row*column);
    putchar('\n');
  }
}
```

The output is a multiplication table of 10x10 size.

Syntax of *while* loop is as follows:

```
while (condition)
   {
      Statement;
   }
```

(example-8)

```
#include<stdio.h>
main()
   {
    int i=0;
    while(i<10)
      {
       printf(" \n%d",i)
       i++;
      }
    }
```

It will print the numbers from 0 to9

Syntax of *do while* loop is as follows:

```
do
{
   statement1;
   statement2;
}
while(condition);
```


(example-9)

```
#include<stdio.h>
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’.
