CONSTANTS IN C PROGRAMMING

In C any literal number, single character or string of characters is known as a constant. Consider the following expression :

```
fTotal = 30 + fAmount;
```

Here fTotal and fAmount are variables and the number 30 is a constant. This value will not change in a program. A user using a program having constants cannot change the constants when the program is executing. Constants are generally hard-coded into a program. Unintended modifications are prevented from occurring. The compiler will catch attempts to reassign new values to constants.

There are three ways for defining a constant in C. First method is by using #define statements. #define statement is a preprocessor directive . It is used to define constants as follows :

```
#define TRUE 1
#define FALSE 0
#define PI 3.1415
```

Wherever the constant appears in a program, the preprocessor replaces it by its value.

The second method is by using the const keyword in conjunction with a variable declaration.

```
const float PI = 3.1415;
```

Here, the compiler will throw an error whenever the variable PI declared as a constant is assigned a new value.

The third method is by using enumerated data type. This data type is initiated by the keyword enum, immediately after the keyword enum is the name of the variable, followed by a list of values enclosed in curly braces. This defines a set of constants. Hence eliminates the usage of multiple #define statements.

```
enum answer {FALSE, TRUE};
```

Thus

```
answer = "No";
```

is an invalid statement.

This example is to show how you would output constant values using printf() routine.

Program 3.4

```
/* Program to show using #define to declare constants */
#include <stdio.h>
#define TRUE 1
#define FALSE 0
#define PI 3.1415
void main()
  {
printf("This is TRUE as an integer: %d \n", TRUE);
printf("This is FALSE as an integer: %d \n", FALSE);
printf("This is PI as an integer: %d \n", PI);
printf("This is PI as a float: %.4 \n", PI);
  }
```

The %c is to tell printf() routine to format the variable f as a character.

In C an enumerator is of type int. Unless specified the first value is internally assigned the value 0. So FALSE is assigned a value 0 and TRUE a 1. For each additional constant in the enumeration the value increases by 1.

enum COLOR {RED, BLUE, GREEN, YELLOW};

Enumerated values that appear subsequently after YELLOW without a specific value will be assigned sequential values with 53 plus one.

Consider another enumerated data type :

enum direction {north, south, east = 40; west};

Here north =0, south=1, east=40 and west=41.

Source : http://www.peoi.org/Courses/Coursesen/cprog/frame3.html