Getting Started with C

IN 1101 PROGRAMMING FUNDAMENTALS

Getting Started with C

Why C?

Variables, Constants and Keywords

- Variable – A data storage location that has a value that can change during program execution.
- Constant – Fixed value that can't change.
- Keyword – A word that carries special meaning.

Understanding Computer Memory

- Computer uses random-access memory (RAM) to store information while it is operating.
- RAM is volatile.
- Each byte of memory in RAM has a unique address.
- Unique address can be used to distinguish the bytes of memory from other bytes.

Variables

- A variable is a named data storage location in your computer's memory.
- Rules for constructing variables names:
  - The name can contain letters (a to z and A to Z), digits (0 to 9), and the underscore character (_).
  - No commas or blanks are allowed within a variable name.
  - The first character in the variable name must be an alphabet or underscore (_).
  - Case matters.
  - General styles to create long variables- Using an underscore to separate words or using camel notation.

e.g.

Variable name	Legality
Sum	✓
4sale	×
savings#account	×
SavingsAccount	✓
Savings_Account	✓

Variable Declarations & Initializing

- Before you use a variable in a C program, it must be declared.
- A variable declaration has the following form :

  ***typename varname*;**

  e.g. int count;

  float percent, total;

- Location of variable declarations in the source code is important. **WHY ?**
- **Initialization:** Before using a variable, you should always initialize it to a known value.
  - Can do this independently of the variable declaration by using an assignment statement.
    - int count; /* Set aside storage space for count */
    - count = 0; /* Store 0 in count */
  - You can also initialize a variable when it's declared

    int count = 0;

Constants

- Constant - value stored in a constant can’t be changed during program execution.

Symbolic Constants

- Symbolic constants are constants which are represented by name or symbol in a C program.
- Defining symbolic constants :
  - Using **#define** directive
    - #define CONSTANTNAME literal
    - E.g. #define PI 3.14159
  - Using **const** keyword
    - E.g. const int count = 100

Literal Constants

- Literal Constant – A value that is types directly into the source code whenever it is needed.
- e.g. const int = 50 ; is a constant integer expression where 50 is an integer literal
- Four types of literals:
  - Integer literals
  - Float literals
  - Character literals
  - String literals

C Keywords

- Keywords are the words whose meaning has already been explained to the C compiler.
- Keywords cannot be used as variable names.

break	double	return
case	else	long
char	for	register
const	float	static
default	if	switch

The first C Program

```
/* Calculation of simple interest */
# include <stdio.h>
int main( )
{
        int p, n ;
        float r, si ;
        p = 1000 ;
        n = 3 ;
        r = 8.5 ;
        /* formula for simple interest */
        si = p * n * r / 100 ;
        printf ( "%f\n" , si ) ;
        return 0 ;
}
```

Comments in C Program

- Comments are used in a C program to clarify either the purpose of the program or the purpose of some statement in the program.

e.g `/* Calculation of simple interest */`

```
/* This comment has
   three lines
   in it */
```

```
// Calculation of simple interest (ANSI C permits this comment)
```

What is main()?

- **main()** is a function.
- **main()** function is an entry point of the programming code to start its execution.
- **main()** is a universally accepted keyword, therefore cannot change its meaning and name.
- Return type is defined before **main().**
  - If main() function returns an integre value -→ int main()
  - If main() function returns nothing → void main()

Variables and their Usage

- Any variable used in the program must be declared before using it.For example,

```
int p, n ;

float r, si ;
```

printf()

- Output to screen is achieved using readymade library functions like ***printf().***
- For us to be able to use the printf( ) function, it is necessary to use **#include <stdio.h>** at the beginning of the program.
  - **#include** is a preprocessor directive.
- The general form of printf( ) function is,

  printf ("<format string>", <list of variables>) ;
  - <format string> can contain,
    - %f for printing real values
    - %d for printing integer values
    - %c for printing character values
  - E.g. printf ("%f", si) ;

    printf ("%d %d %f %f", p, n, r, si) ;

Receiving Input – *scanf()*

- **scanf()** can be used to get user inputs through the keyboard during program execution.

```
scanf ( "%d %d %f", &p, &n, &r ) ;
```

- ampersand (&) before the variables in the scanf( ) function is a must.

- gives the location number (address) used by the variable in memory

Questions ?