SIES College of Arts, Science and Commerce

Class: SYBA Sub: Computer Programming Sem: III Unit: III

Topic: **C Programming**

Q1. What is C?

A1: C is a programming language developed at AT & T's Bell Laboratories of USA in 1972. It was designed and written by a man named Dennis Ritchie.

Q2. What is the role of C Programming Language in Operating Systems?

A2: Major parts of popular operating systems like Windows, UNIX, Linux is still written in C. This is because even today when it comes to performance (speed of execution) nothing beats C. Moreover, if one is to extend the operating system to work with new devices one needs to write device driver programs. These programs are exclusively written in C.

Q3. What are the uses of C Programming Language in electronic devices?

A3: Mobile devices like cellular phones and palmtops are becoming increasingly popular. Also, common consumer devices like microwave oven, washing machines and digital cameras are getting smarter by the day. This smartness comes from a microprocessor, an operating system and a program embedded in these devices. These programs not only have to run fast but also have to work in limited amount of memory. No wonder that such programs are written in C. With these constraints on time and space, C is the language of choice while building such operating systems and programs.

Q4. Explain the analogy between English Language and C Programming Language?

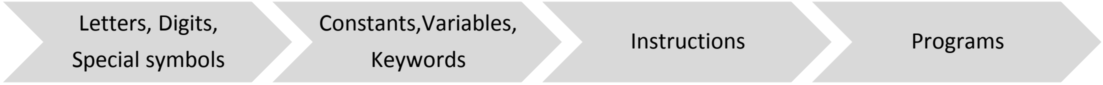
A4: There is a close analogy between learning English language and learning C language. The classical method of learning English is to first learn the letters used in the language, then learn to combine these letters to form words, which in turn are combined to form sentences and sentences are combined to form paragraphs.

Learning C is similar and easier. Instead of straight-away learning how to write programs, we must first know what letters, numbers and special symbols are used in C, then how using them constants, variables and keywords are constructed, and finally how are these combined to form an instruction. A group of instructions would be combined later on to form a program.

English Language

Letters
Words
Sentences
Paragraphs

C Language


Letters, Digits, Special symbols
Constants,Variables, Keywords
Instructions
Programs

Q5. What is a Constant in C? What are it’s different types?

A5: A constant is an entity that doesn’t change. C constants can be divided into two major categories: (a) Primary Constants, (b)Secondary Constants. Primary Constants are further classified into (i) Character Constants, (ii) Integer Constants and (iii) Real constants. Secondary Constants are further classified into (i) Arrays, (ii) Pointers, (iii) Structures, etc.

Q6. What are the rules for constructing Character constants in C?

A6: (1) A character constant is a single letter, a single digit or a single special symbol enclosed within single inverted commas. Both the inverted commas should point to the left.
For example, ’A’ is a valid character constant whereas ‘A’ is not.
(2) The maximum length of a character constant can be 1 character. Examples are : 'A' , 'I' , '5' , '='.

Q7. What are the rules for constructing Integer constants in C?

A7: (1) An integer constant must have at least one digit.
(2)It must not have a decimal point.
(3)It can be either positive or negative.
(4)If no sign precedes an integer constant it is assumed to be positive.
(5)No commas or blanks are allowed within an integer constant.
(6)The allowable range for integer constants is -32768 to 32767.
Examples: 426, +782, -8000, -7605.

Q8. What are the rules for constructing Real constants in C?

A8: (1)A real constant must have at least one digit.
(2)It must have a decimal point.
(3)It could be either positive or negative.
(4)Default sign is positive.
(5)No commas or blanks are allowed within a real constant.
Examples: +325.34, 426.0, -32.76, -48.5792.

Q9. What is a Variable in C? What are it’s different types?

A9: An entity that may vary during program execution is called a variable. Variables are the names given to locations in a computer memory. These locations can contain integer, real or character constants. The types of variables depend on the types of constants that the variables can handle. This is because a particular type of variable can hold only the same type of constant. For example, an integer variable can hold only an integer constant, a real variable can hold only a real constant and a character variable can hold only a character constant.

Q10. What are the rules for constructing Variables in C?

A10: (1) A variable is any combination of 1 to 31 letters, digits or underscores.
(2) The first character in the variable must be a letter or an underscore.
(3) No commas or blanks are allowed within a variable.
(4) No special symbol other than an underscore can be used in a variable.
Examples: si_int, m_hra, pop_e_89.

Q11. What is a C keyword?

A11: A C keyword is a word whose meaning has already been explained to the C compiler (or in a broad sense to the computer). The keywords cannot be used as variable names because if we do so we are trying to assign a new meaning to the keyword, which is not allowed by the computer. The keywords are also called 'Reserved words'. There are only 32 keywords available in C. They are auto, double, int, struct, break, else, long, switch, case, enum, register, typedef, char, extern, return, union, const, float, short, unsigned, continue, for, signed, void, default, goto, sizeof, volatile, do, if, static, while.

Q12. What are the rules for writing a C program?

A12: (i) Each instruction in a C program is written as a separate statement. Therefore a complete C program would comprise of a series of statements.
(ii) The statements in a program must appear in the same order in which we wish them to be executed; unless of course the logic of the problem demands a deliberate 'jump' or transfer of control to a statement, which is out of sequence.
(iii) Blank spaces may be inserted between two words to improve the readability of the statement. However, no blank spaces are allowed within a variable, constant or keyword.
(iv) All statements are entered in small case letters.
(v) C has no specific rules for the position at which a statement is to be written. That's why it is often called a free-form language.
(vi) Every C statement must end with a semicolon(;). Thus ; acts as a statement terminator.

Q13. What is a type declaration statement in a C program?

A13: Type declaration in a C program is a statement that specifies the data-type of the objects (eg: variables) to be used in the program. For example,
(i) int x;
This specifies that x is an integer type variable, i.e., x is the name of a memory location which can save only an integer constant.

(ii) float y;

This specifies that y is a floating point type or real type variable, i.e., y is the name of a memory location which can save only a real constant.

(iii)char z;

This specifies that z is a character type variable, i.e., z is the name of a memory location which can save only a character constant.

Q14. What is a format specifier in C? List any 3 format specifiers in C.

A14: Format specifier in C is a combination of the % sign with a suitable single character and is used to specify the type of the variable in use within a C function. A few format specifiers in C are

(i) %d used to specify an integer type variable

(ii) %f used to specify a real type(floating point type) variable

(iii) %c used to specify a character type variable.

Q15. Name any five functions used in a C program and give their uses.

A15: The following is the list of the names and uses of four C functions:

(i) main( )

This function encloses the entire body of the program within a pair of braces { }. In cases where the program does not return any value, the keyword void is placed before main( ) as void main( ).

(ii) printf( )

This function is used to display the output on the computer screen(console).

The syntax is printf ("<format string>", <list of variables>) ;

Here format string means the statement that comprises of format specifiers and possibly other characters.

(iii) scanf( )

This function is used to store a constant into a variable.

The syntax is scanf(“<list of format specifiers>”, &variable1, &variable2…)

Eg: scanf(“%d %f %c”, &x, &y, &z) will store an integer constant in the variable x, a real constant in the variable y and a character constant in the variable z.

(iv) getch( )

This function will make the console(output screen) wait for the user to enter a character.

(v) clrscr( )

This function will clear the last produced output on the console(the output screen)

Q16. What is an escape sequence in C? List any three of them with their uses.

A16: An escape sequence is a combination of the backslash \ with a character and is used in printf( ) to help the control to escape through big spaces or tabs or even lines.

A few of them along with their uses are as follows:

(i) \n for new line
(ii) \t for single horizontal tab space
(iii) \v for single vertical tab space.

Q17. What is a comment and its uses in C?

A17: Sometimes it is not so obvious as to what a particular statement in a program accomplishes. At such times it is worthwhile mentioning the purpose of the statement (or a set of statements) using a comment. A comment in C is a statement enclosed within /*   */ which is written just to mention the purpose of the program or a program statement.

Q18. What are C header files? Name any two.

A18: A header file is a file with extension .h which contains C function declarations and macro definitions. Two of them are stdio.h and conio.h. They have to be written before main() function prefixing the word include as follows:

```
#include<stdio.h>
#include<conio.h>
```

Q19. What are the commands/steps to perform the following in C?
1. To open a blank window
2. To save a program
3. To compile a program
4. To run a program

A19:
1. File Menu → New
2. File Menu → Save As → Type 'C:\TC\BIN\program_name.C'→ OK
3. ALT F9
4. CTRL F9

Q20. Write a C program

i. To print 'I am a student of SYBA Computer Programming'

Prog:
```
#include<stdio.h>

void main( )
{
      printf("I am a student of SYBA Computer Programming");
      getch( );
}
```

ii. To accept two integers and print their sum.

Prog:
```
#include<stdio.h>
#include<conio.h>

void main( )
```

```
{
      int x, y, s;       /* Type Declaration*/

      printf(“Enter the first integer\n”);
      scanf(“%d”, &x);
      printf(“Enter the second integer\n”);
      scanf(“%d”, &y);

      s = x+y;
      printf(“The sum is %d”, s);
      getch( );
}
```

iii. To accept two real numbers and print their product(result of multiplication).

Prog:

```
#include<stdio.h>
#include<conio.h>

void main( )
{
      float x, y, p;       /* Type Declaration*/

      printf(“Enter the first real number\n”);
      scanf(“%f”, &x);
      printf(“Enter the second real number\n”);
      scanf(“%f”, &y);

      p = x * y;
      printf(“The Product is %f”, p);
      getch( );
}
```

iv. To receive a character from the user and print it.

Prog:

```
#include<stdio.h>
#include<conio.h>

void main( )
{
      char x;
      clrscr( );

      printf(“Enter any character\n”);
      scanf(“%c”, &x);
```

```
    printf(“The character you have entered is %c”, x);
    getch( );
}
```

v. To find the simple interest after accepting the principal, period and rate of interest from the user(formula: I = P*N*R/100).

Prog:

```
#include<stdio.h>
#include<conio.h>

void main( )
{
    float P, R, I;
    int N;
    clrscr( );

    printf(“Enter the value of Principal\n”);
    scanf(“%f”, &P);
    printf(“Enter the number of years\n”);
    scanf(“%d”, &N);
    printf(“Enter the rate of interest\n”);
    scanf(“%f”, &R);

    I=P*N*R/100;
    printf(“The simple interest is %f”, I);
    getch( );
}
```

Q21. What is a Decision Control Instruction? Describe any two.

A21: Many a times, we want a set of instructions to be executed in one situation, and an entirely different set of instructions to be executed in another situation. This kind of situation is dealt in C programs using a decision control instruction.

Two such types are as follows:

(i) *if*

```
Syntax: if(condition)
                statement;
```

Use: This type of decision control instruction executes the statement if the condition within the parentheses is true

(ii) *if-else*

```
Syntax: if(condition)
                statement1;
            else
                statement2;
```

Use: This type of decision control instruction executes the statement1 if the condition within the parentheses is true and executes the statement2 if the condition is false.

Q22. Write a C Program

i. To accept a character from the user and print WOW if the character is $.

Prog:

```
#include<stdio.h>
#include<conio.h>

void main( )
{
    char  x;
    clrscr();

    printf(“Enter any character\n”);
    scanf(“%c”, &x);

    if(x==’$’)
        printf(“WOW”);
    getch( );
}
```

ii. To accept marks(out of 100) and print PASSED if the marks are more than or equal to 40 and FAILED otherwise.

Prog:

```
#include<stdio.h>
#include<conio.h>

void main( )
{
    float m;
    clrscr( );

    printf(“Enter your marks(out of 100)\n”);
    scanf(“%f”, &m);

    if(m >= 40)
        printf(“PASSED”);
    else
        printf(“FAILED”);
    getch( );
}
```

iii. To receive the rate and quantity of a product purchased and to display the final price. There is a 10% discount to be offered if the cost of the product purchased exceeds Rs. 1000.

Prog:

```
#include<stdio.h>
#include<conio.h>

void main( )
{
      float r, qty, price, fin_price;
      clrscr( );

      printf(“Enter the rate of the product\n”);
      scanf(“ %f”, &r);
      printf(“Enter the quantity of the product\n”);
      scanf(“ %f”, &qty);

      price=r * qty;

      if(price > 1000)
            printf(“The final price is %f”, price - 0.1*price);
      else
            printf(“The final price is %f”, price);

      getch( );
}
```

Q23. What is a Loop Control Instruction? Describe any one of them.

A23: The versatility of the computer lies in its ability to perform a set of instructions repeatedly. This involves repeating some portion of the program either a specified number of times or until a particular condition is being satisfied. This repetitive operation is done through a loop control instruction. One of the loop control instruction is:

The *for* loop

```
Syntax: for( initialise counter ; test counter ; increment counter )
              {
                    statement1;
                    statement2;
                    :
                    :
              }
Example: for(i=1; i<5; i++)
               {
                  printf(“This is CP\n”);
                }
```

This will print This is CP 4 times on the console one below the other.

Q24. Write a C program

i. To print Programming 5 times with 1 tab space between every pair of words.

Prog:

```
#include<stdio.h>

void main( )
{
      int i ;
      clrscr( );

      for(i=1; i<6; i++)
      {
            printf(“Programming\t”);
      }
      getch( );
}
```

ii. To find and display the sum of first 10 natural numbers(i.e. to find 1+2+…..+10)

Prog:

```
#include<stdio.h>

void main( )
{
      int i , sum;
      clrscr( );

      sum=0;
      for(i=1; i<11; i++)
      {
            sum = sum + i;
      }
      printf(“The sum of first 10 natural numbers is %d”, sum);
      getch( );
}
```

iii. To find and display the product of first 10 natural numbers(i.e. to find 1*2*…..*10)

Prog:

```
#include<stdio.h>

void main( )
{
      int i , prod;
      clrscr( );
```

```
    prod = 1;
    for(i=1; i<11; i++)
    {
        prod = prod * i;
    }
    printf(“The product of first 10 natural numbers is %d”, prod);
    getch( );
}
```

iv. To find the 10th term of an Arithmetic Progression after receiving the 1st term and the common difference.(Arithmetic Progression is a sequence of numbers in which the difference between a term and its adjacent term is constant)

Prog:

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int i ;
    float a, d, t;
    clrscr( );

    printf(“Enter the first term\n”);
    scanf(“%f”, &a);
    printf(“Enter the common difference\n”);
    scanf(“%f”, &d);

    t = a;
    for(i=1; i<10; i++)
    {
        t = t + d;
    }
    printf(“The 10th term of the Arithmetic Progression is %f”, t);
    getch( );
}
```