UNIT-2

22. Explain how to declare a variables with examples?

Syntax of declaring a variable is:-

datatype variablename;

or

datatype var1,var2,……..varn;

Example:-

1) Declare an integer Variable int sum;
2) Declaring Floating point variable float avg;
3) Declare Character Variable char ch;
4) Declare Double Precision Variable double d;
5) Declare Multiple Integer Variable in single declaration statement

int a1,a2,a3,a4;

23. Explain how to assaign values to variables?

We can assaign a value to a variable in two ways:-

- Using Assignment Operator
- Reading Data from Keyboard

Using Assignment Operator:-

Syntax:-data_typevariable_name=constant;

Examples:-

1)int x=10; or int x;
x=10;

2)int x,y,z; or int x=y=z=10;
x=y=z=10;

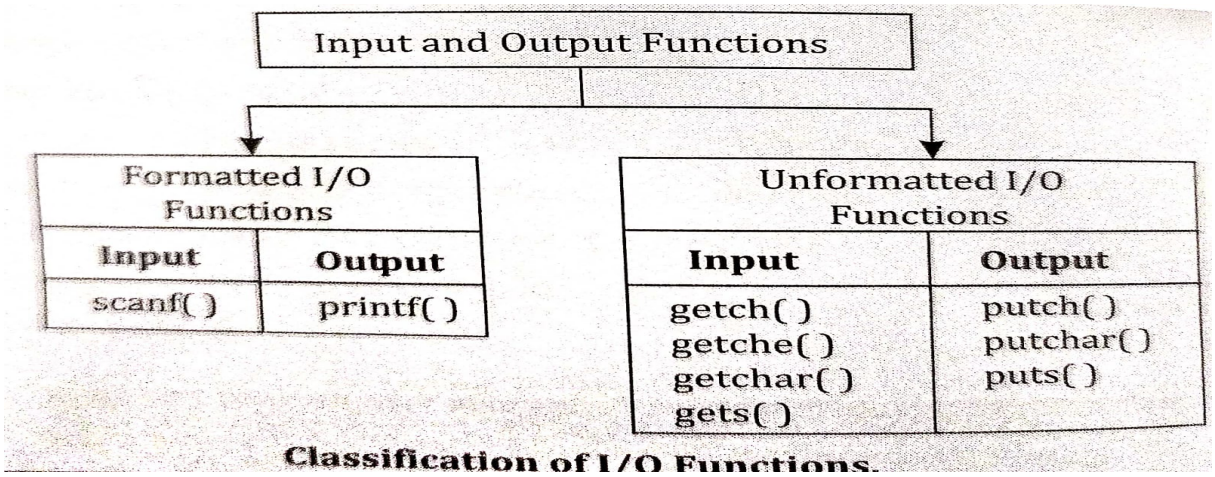
24. Write a C program to illustrate Declaration and Assignments in Variables?

```
#include<stdio.h>
void main()
{
  /*  ------ only declaration ---------- */
  int x,y;
```

```
float p,q;
/* ------- Declaration and Assignments ---------- */
int z=20;
float r= 3.142;
/* ----------- only Assignments ------------------ */
x = y = 10;
p = q = 1.0;
/* --------- Display ---------------------- * /
printf("x=%d\n",x);
printf("y=%d\n",y);
printf("z=%d\n",z);
printf("p=%d\n",p);
printf("q=%d\n",q);
printf("r=%d\n",r);
}
```

25. What are the Console input/output functions? Mention its types?

- **Console Input/output Functions**: Console Input Output functions help to read data from Keyboard and Print/ Display data to the Monitor or screen.
- **Types**:-
  1) Formatted Input/Output Functions.
  2) Unformatted Input/Output Functions.



Classification of I/O Functions.

26. Explain Formatted Input/Output Functions?

- Formatted Input/output Functions that allow input and output operations to be performed in a specified and desired format.

OR

- Formatted I/O functions are used to take various inputs from the user and display multiple outputs to the user.
- These types of I/O functions can help to display the output to the user in different formats using the format specifiers. These I/O supports all data types like int, float, char, and many more.
- **Example:** The Formatted I/O Functions are scanf() and printf().
- The printf() is used to display the formatted data items on the standard output device normally the monitor.
- The scanf() is used to read the formatted data input from the standard input device normally the keyboard.
- These functions are defined in the header file stdio.h and return EOF if there occurs an error or end of file.

27. Explain scanf() function with examples?

- The scanf() function is the formatted input function.
- The scanf() is used to accept the values of any data type.
- This function scans or accept the values from the standard input device(keyboard) to the address of variables mentioned in the argument list.
- **Syntax:-**
  scanf("format string", &var1, &var2,……. &varn);

Note:-
The ampersand symbol & before variable name is an operator that

specifies the variable name's address.

Examples for scanf():-

1) int a;

scanf("%d", &a);

2) char ch;

6) int a, b, c;

scanf("%d %d %d", &a, &b, &c);

7) int dd ,mm, yy;

scanf(“%c”, &ch);

3)float f;

scanf(“%f”, &f);

4)double d;

scanf(“%lf”, &d);

5) int a;

float f;

double d;

scanf(“%d %f %lf ”, &a, &f, &d);

scanf(“%d-%d-%d”, &dd, &mm, &yy);

8) int a, b, c, v;

v=scanf(“%d %d %d”, &a , &b, &c);

Example:-

Program to illustrate the return value of scanf():-

```
#include<stdio.h>
#include<conio.h>
main()
{
   int a ,b, c, v;
printf(“Enter the values of a, b and c \n”);
   v= scanf(“%d %d %d”, &a, &b, &c);
printf(“\n Number of values successfully read is: %d”, v);
getch();
}
```

Output:-

Enter values of a, b and c

10 20 30

Number of values successfully read is: 3

28. Explain printf() function with examples?

- The printf() function is the formatted output function.
- This function prints all type of data values to the console.

- **Syntax:-**
  printf(“ format string”, list of variables);

Where format string consists of

i) Characters that are simply printed as they are.
ii) Format Specifiers that begin with %sign. Example: %d, %f, %s etc.
iii) Escape Sequence characters that begin with backslash(\) character. Example: \n, \t, \b etc.

- The **list of variables** are the variables whose values are formatted and printed according to the specification of the format string. The list of variablesshould match in number, order and type with the format specification.

Example:-

```
  #include<stdio.h>
  #include<conio.h>
  void main()
{
int  a=10;
  float f=3.142;
  double d= 8.525;
   char ch = ‘A’
printf(“%d %f  %lf %c”, a, f, d, ch);
}
```

Output:-

10 3.142 8.525 A

29. Explain Format Specifiers in C language?

- Format Specifiers specify the type of data that can be displayed from the variables corresponding to the list of variables with the required format.
- This should match with the variables in numbers, order and type.
- The %d, %f ,%lf , %s and %c are called Format Specifiers . These are also called conversion symbols or conversion specifications.

- C Format Specifiers are used in input/output functions to display data in a specified format so that it can be read easily and printed in a desired format.

30. Explain Escape sequence in C language?

- Certain non printable characters which are used in the printf() functions are called as Escape Sequences.
- An Escape sequence begins with a backslash '\' followed by a character or characters.
- Escape sequence are used to format the output text and are not generally displayed on the screen.
- Each escape sequence has its own predefined function.
- Example:-

\n is used to create a new line and place the cursor in the beginning of new line. Words come after '\n' will be pushed to new line.

Escape Sequence	Meaning
\n	New line
\b	Back space
\f	Form feed
\'	Single quote
\t	Horizontal Tab
\a	bell / alert
\r	Carriage return
\v	Vertical tab
\"	Double quote
\0	NULL

Examples:-

```
1)main()
    {
printf("Programming in C \n");
printf("For Beginners");
    }
```

Output:-

Programming in C

For Beginners

```
2)main()
      {
printf("Programming in C \t");
printf("For Beginners");
      }
```

Output:-

Programming in C For Beginners

31. Explain Field Width Specification for printing Integers with Examples?

- Field width can be specified in the format specification.
- The printf function uses this field width to know how many columns on the screen should be used while printing a value.
- Syntax:-

%WC right justified

%-WC left justified

where

W indicates total number of columns required to print the value. It is also called the total width of the output.

C is the format specifiers for the integers. C can be one of the format specifiers for integers like %d, %u , %lu, %x, %o etc.

Displaying the value from right to left is said to be **right justified** and Displaying the value from left to right is said to be **left justified.**

Example:-

Statement	Output
printf ("%d", 1234);	1 2 3 4
printf ("%5d", 1234);	1 2 3 4
printf ("%8d", 1234);	1 2 3 4
printf ("%2d", 1234);	1 2 3 4 Additional columns are created automatically.
printf ("%–5d", 1234);	1 2 3 4
printf("%06d", 1234);	0 0 1 2 3 4 The leading blanks will be filled with Zeros by placing the 0 before the field width specifier.
printf("%d",9876543210)	This is very long number and hence output will be junk value. Use %ld for long numbers instead of %d
printf("%ld",9876543210)	9 8 7 6 5 4 3 2 1 0

32. Explain Field Width Specification for printing Decimal numbers or float with Examples?

- The syntax of specifying field width for float values is shown below.
- **Syntax:-**

%W.Xf right justified
%-W.Xf left justified

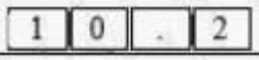
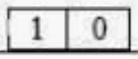
where

W indicates total number of columns required to print the value. It is also called the total width of the output.

X is the number of columns used after the decimal part.

f is the format specifier for float data type.

Example:-

Input	Statement	Output
n=10.20	printf("%7.2f",n); **Note:** Print in right justification	
n=10.20	printf("%-7.2f",n); **Note:** Print in left justification	
n=10.20	printf("%2.2f",n);	
n=10.20	printf("%2.3f",n);	
n=10.20	printf("%1.3f",n);	
n=10.20	printf("%1.1f",n);	
n=10.20	printf("%1.0f",n);	
n=10.20	printf("%0.3f",n);	
n=10.20	printf("%0.2f",n);	

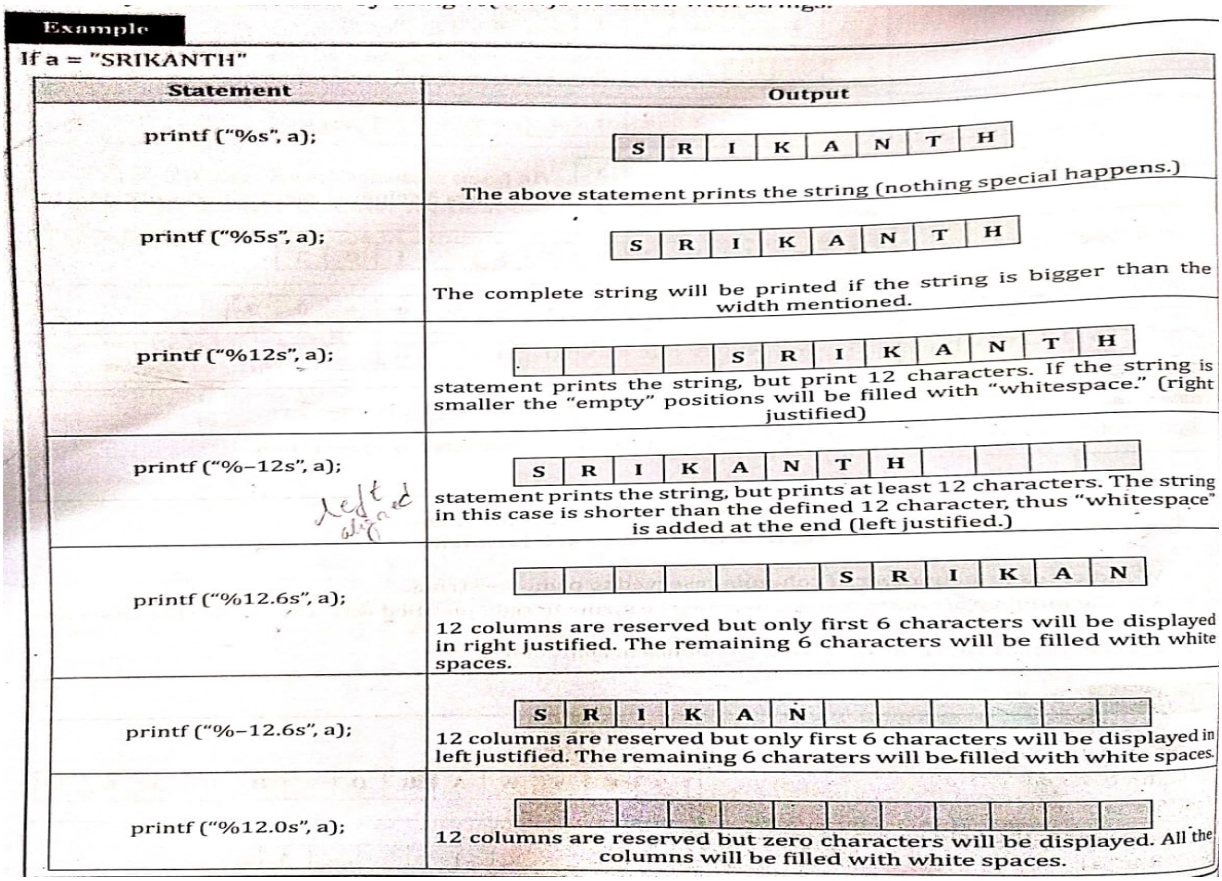
33. Explain Field Width Specification for printing Character and string with Examples?

- The syntax of specifying field width for strings and single characters is shown below.
- **Syntax:-**

%W.Xs right justified
%-W.Xs left justified

where
W indicates total number of columns reserved to print the string.
X is the number of columns used to print the string in right justified way. The rest of characters are ignored.
s is the format specifier for string.

Example:-



Example

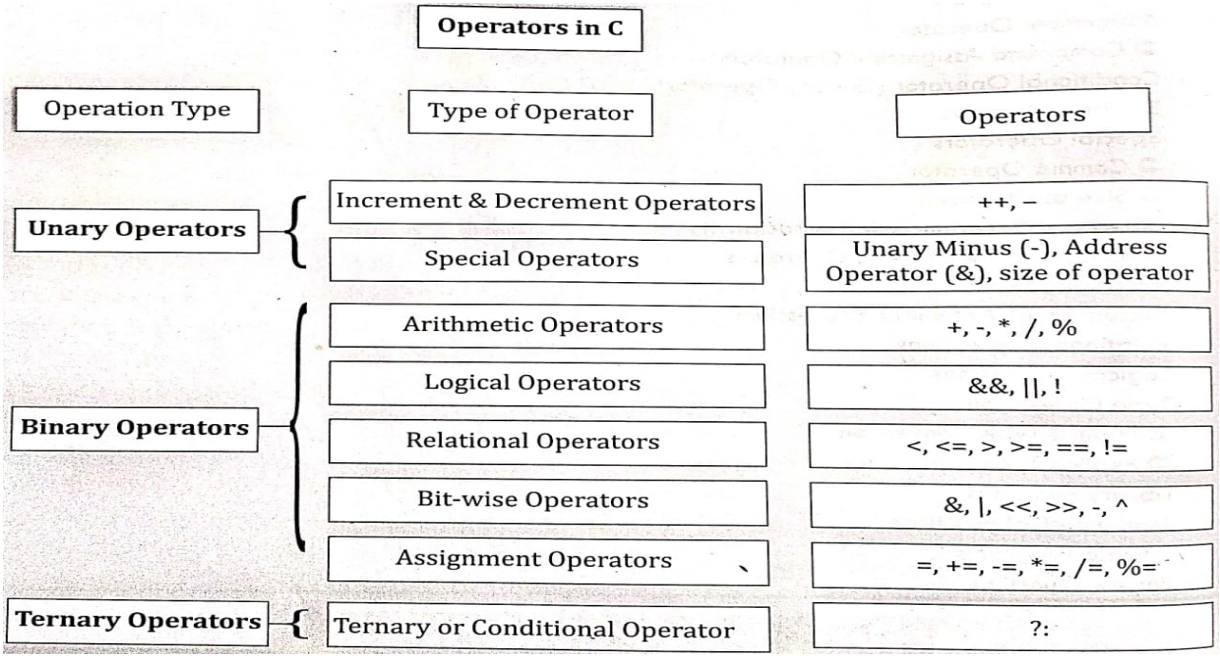
If a = "SRIKANTH"

Statement	Output
printf ("%s", a);	S R I K A N T H The above statement prints the string (nothing special happens.)
printf ("%5s", a);	S R I K A N T H The complete string will be printed if the string is bigger than the width mentioned.
printf ("%12s", a);	S R I K A N T H statement prints the string, but print 12 characters. If the string is smaller the "empty" positions will be filled with "whitespace." (right justified)
printf ("%–12s", a);	S R I K A N T H statement prints the string, but prints at least 12 characters. The string in this case is shorter than the defined 12 character, thus "whitespace" is added at the end (left justified.)
printf ("%12.6s", a);	S R I K A N 12 columns are reserved but only first 6 characters will be displayed in right justified. The remaining 6 characters will be filled with white spaces.
printf ("%–12.6s", a);	S R I K A N 12 columns are reserved but only first 6 characters will be displayed in left justified. The remaining 6 charaters will be filled with white spaces.
printf ("%12.0s", a);	12 columns are reserved but zero characters will be displayed. All the columns will be filled with white spaces.

34. Explain operators in C language with examples?

- **Operators** are symbols that represent operations to be performed on one or more operands.
- **Unary operator** is a single operator that operates on a single operand to produce a new value. Unary operators can perform operations such as negation, increment, decrement
- A **Binary operator** is an operator that operates on two operands to produce a new value (result). Most common binary operators are +, -, *, /, etc.
- **Ternary Operator** in C is an operator which takes three operands or variables, unlike the other operators which take one or two operands. Ternary operator in C is also known as the Conditional Operator.

- **The increment(++) and decrement operators(--)** are important unary operators in C. The increment operator increases the value of the variable by one and the decrement operator decreases the value of the variable by one.



- **Special operators** are used to perform specific operations that cannot be done with normal arithmetic or logical operators.
- **Arithmetic operators** are used to perform common mathematical operations.
- **Assignment operators** are used to assign values to variables.
- **Relational operators** are used to compare two values (or variables). This is important in programming, because it helps us to find answers and make decisions.
- **logical operators** are used test true or false values.
- **The bitwise operators** are the operators used to perform the operations on the data at the bit-level.
- The **conditional operator** contains a condition followed by two statements or values.
- **Syntax** of Conditional/Ternary Operator in C
  variable = Expression1 **?** Expression2 **:** Expression3**;**

Example:-

```
#include <stdio.h>
int main()
{
  int a=5, b;
  b=((a==5)?(3):(2)); // conditional operator
  printf("The value of 'b' variable is : %d",b);
   return 0;
}
```

35. What are the Arithmetic Expressions? Explain its types?

- A group of constants, variables and operators arranged as per the syntax of the language is referred as Arithmetic expression.

- Arithmetic expression can be classified into 3 types:-

1) **Integer Arithmetic Expression: I**n an expression, if all the values are integers then the expression is called as Integer Arithmetic Expression.

Integer Expression	Evaluated Value
5 / 2	2
5 * 2	10
(5 + 8) / 2	6
8 - 3 * 2	2
-100 + 8	-92

2) **Real Arithmetic Expression: I**n an expression, if all the values are in float form then the expression is called as Real Arithmetic Expression.

Real Expressions	Evaluated value
5.0/2.0	2.5
2.0+10.5	12.5
10.5+2.0*1.0	12.5
3.0 – 10.5	–6.5
2.0 – (–9.5)	11.5

3) Mixed Arithmetic Expression: when an expression is composed of int and float then the expression is called as Mixed Arithmetic Expression.

Mixed Mode Expressions	Evaluated Value
5.0/2	2.5
2.0/5	0.4
3.5+4	7.5
2.5*4/2	5.0
2*(1.5*2.5)	8.0