Unit no. 02: User Interactions

Briefly answer the following questions:

1. **What is the difference between *scanf* and *getch?***

scanf:

- Purpose: Reads formatted input from the standard input (keyboard).
- Header File: Requires #include <stdio.h>
- Usage: Used to input integers, floats, characters, strings, etc.
- Example:
  int num;
  scanf("%d", &num);

getch:

- Purpose: Gets a single character from the keyboard *without echoing it* to the screen.
- Header File: Requires #include <conio.h>
- Usage: Often used to pause the screen or wait for a key press.
- Example:
  getch(); Waits for a key press, does not display the key

2. **Which function of C language is used to display output on screen?**

The function used in C language to display output on the screen is:

printf()

- Purpose: Displays formatted output to the standard output (usually the screen).
- Header File: **#include <stdio.h>**
- Syntax:

  printf("message to display"); *// For simple use*

  printf ("format specifier", variable);

Example:

```
#include <stdio.h>
int main()
{
   int age = 20;
   printf("My age is %d", age);
   return 0;
}
```

Output:

My age is 20

3. Why are format specifiers important to be specified in I/O operations?

Format specifiers tell the compiler what type of data is being input or output during scanf() or printf () operations. Without format specifiers, the compiler wouldn't know whether you're dealing with an int, float, char, or string. Different data types occupy different amounts of memory. Format specifiers help the compiler read/write the correct number of bytes.

Using the wrong specifier (e.g., %d for a float) can cause unpredictable behavior or incorrect outputs.

4. What are escape sequences? Why do we need them?

Escape sequences are special characters in C used to perform specific formatting tasks in strings or output. They begin with a backslash (\), followed by a character.

We need escape sequences to:

- **Format output** (e.g., move to a new line, add a tab space)
- **Print special characters** that can't be typed directly (like quotes, backslashes)
- **Control screen output** behavior in a readable and structured way

Common Escape Sequences:

Escape	Meaning
\n	New line
\t	Horizontal tab

5. Which operators are used for arithmetic operations?

Arithmetic Operators in C

Arithmetic operators are used to perform basic mathematical operations like addition, subtraction, multiplication, etc.

List of Arithmetic Operators:

Operator	Meaning	Example (a = 10, b = 5)	Result
+	Addition	a + b	15
-	Subtraction	a - b	5
*	Multiplication	a * b	50
/	Division	a / b	2
%	Modulus	a % b	0

- / Performs integer division if both operands are integers (e.g., 7 / 2 = 3).
- % returns the remainder after division (e.g., 7 % 2 = 1).

6. What are relational operators? Describe with an example?

Relational operators are used to compare two values or expressions. They compare values to determine the relationship between them.

They return a Boolean result:

- 1 (true) if the condition is correct
- 0 (false) if the condition is not met

List of Relational Operators:

Operator	Meaning	Example (a = 10, b = 5)	Result

==	Equal to	a == b	0
!=	Not equal to	a != b	1
>	Greater than	a > b	1
<	Less than	a < b	0
>=	Greater than or equal	a >= b	1
<=	Less than or equal	a <= b	0

7. What are logical operators? Describe with an example?

Logical operators are used to **combine or invert** the results of relational expressions. They help in making **decisions** in conditions like if, while, and for.

List of Logical Operators:

Operator	Name	Description
&&	Logical AND	True if **both** conditions are true
||	Logical OR	False if b**oth** conditions are false
!	Logical NOT	**Inverts** the result (true ↔ false)

8. What is the difference between unary operators and binary operators?

Difference Between Unary and Binary Operators in C

Aspect	Unary Operators	Binary Operators
Definition	Operators that act on **one operand**	Operators that act on **two operands**
Number of Operands	One	Two
Purpose	Usually used for **modifying or testing** a single value	Used to perform **operations between two values**
Examples	++a, --b, -x	a + b, x - y, m * n, p == q
Common Use Cases	Increment, Decrement, Negation, Logical NOT	Arithmetic, Comparison, Logical AND/OR

9. What is the difference between ==operator and = operator?

Difference Between == Operator and = Operator in C

Operator	Name	Purpose	Example	Explanation
=	Assignment Operator	Assigns value to a variable	a = 5;	Assigns value 5 to variable a
==	Equality Operator	Compares two values for equality	a == 5	Returns true if a is equal to 5

10. What is meant by the precedence of operators? Which operator has the highest precedence in C language?

Operator precedence refers to the order in which operators are evaluated in an expression. Operators with higher precedence are evaluated first, while those with lower precedence are evaluated later. In cases where operators have the same precedence, associativity (whether the operators are evaluated from left to right or right to left) decides the evaluation order.

Precedence Rules:

1. Higher precedence operators are evaluated first.
2. Operators with equal precedence are evaluated based on associativity.

Operator:	Precedence:
()	1
!	2
*, /, %	3
+, -	4
> , < , >= , <=	5
== , !=	6
&&	7
||	8
=	9