Conditions and Loops Assignment Questions

Q1. What are conditional statements? Explain conditional statements with syntax and examples.

Solution : Conditional statements are used in programming to make decisions based on certain conditions. These statements execute a block of code only if a specified condition is true. If the condition is false, different code can be executed, or the program may skip the block entirely.

Types of Conditional Statements

Most programming languages provide the following types of conditional statements:

1. if Statement
2. if-else Statement
3. if-elif-else Statement (in Python) or if-else-if Ladder (in Java, C, etc.)
4. Nested if Statement
5. Switch Statement (Available in languages like C, Java, JavaScript, etc.)

1. if Statement

Syntax: if (condition) { // Code to execute if condition is true }

2. if-else Statement : if (condition) { // Code to execute if condition is true } else { // Code to execute if condition is false }

Q2. Write a program that grades students based on their marks

If greater than 90 then A Grad If between 70 and 90 then a B grade If between 50 and 70 then a C grad Below 50 then an F grade

Q3. What are loops, and what do we need them? Explain different types of loops with their syntax and examples?

Solution : Loops are used in programming to execute a block of code multiple times based on a condition. They help in reducing redundancy and automating repetitive tasks efficiently**.** Why Do We Need Loops?

1. Automate Repetitive Tasks – Instead of writing the same code multiple times, loops execute it repeatedly.
2. Iterate Over Data Structures – Loops help in processing arrays, objects, and other collections.
3. Efficient Code Execution – Loops reduce unnecessary code duplication, making programs cleaner and more maintainable.

Types of Loops in JavaScript

1. **for** Loop
2. while Loop
3. do...while Loop
4. for...in Loop (for objects)
5. for...of Loop (for arrays & iterables)

1. for Loop

The `for` loop is used when we **know the number of iterations** in advance. **Syntax:**

for (initialization; condition; increment/decrement) { // Code to execute }

Example: for (let i = 1; i <= 5; i++) { console.log("Iteration:", i); }

2. while Loop

The `while` loop executes as long as the condition is true. It is used when we don't know the number of iterations in advance. For Example:

let count = 1; while (count <= 5) { console.log("Count:", count); count++; }

3. do...while Loop

The `do...while` loop **executes at least once**, even if the condition is false. For example :

let num = 1; do { console.log("Number:", num); num++; } while (num <= 5);

Q4. Generate numbers between any 2 given numbers. Ex

const num1 = 10
const num2 = 25;
Output: 11, 12, 13,
...., 25

Q5. Use the while loop to print numbers from 1 to 25 in ascending and descending order?

Solution :