# Condition & Loops Assignment

## Q 1 . What are Conditional Statements?

**Conditional statements** are programming constructs that allow you to execute specific blocks of code based on certain conditions. They help control the flow of a program by making decisions.

**Syntax:**

- **if statement:**

```
if (condition) {
    // code to execute if condition is true
}
```

- **if-else statement:**

```
if (condition) {
    // code to execute if condition is true
} else {
    // code to execute if condition is false
}
```

- **switch statement:**

```
switch (expression) {
    case value1:
        // code to execute if expression matches value1
        break;
    case value2:
        // code to execute if expression matches value2
        break;
    default:
        // code to execute if no cases match
}
```

**Example:**

```
let score = 85;

if (score > 90) {
    console.log("A Grade");
} else if (score >= 70) {
    console.log("B Grade");
} else if (score >= 50) {
    console.log("C Grade");
} else {
    console.log("F Grade");
}
```

---

## Q 2. Program to Grade Students Based on Their Marks

Here's a JavaScript program that grades students based on their marks:

```
function gradeStudent(marks) {
    if (marks > 90) {
        return 'A Grade';
    } else if (marks >= 70) {
        return 'B Grade';
    } else if (marks >= 50) {
        return 'C Grade';
    } else {
        return 'F Grade';
    }
}

// Example usage:
const marks = 85; // Change this value to test
console.log(gradeStudent(marks)); // Output: B Grade
```

---

## Q 3. What are Loops, and What Do We Need Them?

**Loops** are programming constructs that repeat a block of code multiple times until a specified condition is met. They are essential for automating repetitive tasks and managing collections of data.

**Types of Loops:**

1. **for loop:**
   **Syntax:**

```
for (initialization; condition; increment) {
    // code to execute
}
```

   **Example:**

```
for (let i = 1; i <= 5; i++) {
    console.log(i); // Output: 1, 2, 3, 4, 5
}
```

2. **while loop:**
   **Syntax:**

```
while (condition) {
    // code to execute
}
```

   **Example:**

```
let i = 1;
while (i <= 5) {
    console.log(i); // Output: 1, 2, 3, 4, 5
    i++;
}
```

3. **do-while loop:**
   **Syntax:**

```
do {
    // code to execute
} while (condition);
```

   **Example:**

```
let i = 1;
do {
    console.log(i); // Output: 1, 2, 3, 4, 5
```

```
    i++;
} while (i <= 5);
```

---

## Q 4. Generate Numbers Between Any Two Given Numbers

Here's a JavaScript program to generate numbers between two given numbers:

```
const num1 = 10;
const num2 = 25;

for (let i = num1 + 1; i <= num2; i++) {
    process.stdout.write(i + (i < num2 ? ", " : ""));
}
// Output: 11, 12, 13, ..., 25
```

---

## Q 5. Use the While Loop to Print Numbers from 1 to 25 in Ascending and Descending Order

**Ascending Order:**

```
let i = 1;
while (i <= 25) {
    console.log(i); // Output: 1 to 25
    i++;
}
```

**Descending Order:**

```
let j = 25;
while (j >= 1) {
    console.log(j); // Output: 25 to 1
    j--;
}
```