ASSIGNMENT

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

Ans. Conditional statements are fundamental constructs in programming that allow the execution of different code blocks based on whether a specified condition evaluates to true or false. In other words, they enable a program to make decisions and choose different paths of execution.

The most common types of conditional statements are:

* **if statement**: It executes a block of code if a specified condition is true.

Example: let x = 10;

if (x > 5) {

console.log("x is greater than 5");

}

* **if-else statement**: It executes one block of code if the specified condition is true and another block if the condition is false.

Example: let x = 3;

if (x > 5) {

console.log("x is greater than 5");

} else {

console.log("x is not greater than 5");

}

* **if-else if-else statement** (also known as switch statement in some languages): It allows for multiple conditions to be evaluated sequentially, executing the block of code associated with the first condition that evaluates to true, or a default block if none of the conditions are true.

Example: let x = 10;

if (x > 15) {

console.log("x is greater than 15");

} else if (x > 5) {

console.log("x is greater than 5 but less than or equal to 15");

} else {

console.log("x is less than or equal to 5");

}

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

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

Ans. function gradestudents(**marks**) {

**if** (marks **>** 90) {

**return** 'A grade'**;**

} **else** **if** (marks **>=** 70 **&&** marks **<=** 90) {

**return** 'B grade'

} **else** **if** (marks **>=** 50 **&&** marks **<=** 70) {

**return** 'C grade'

} **else** {

**return** 'F grade '

}

}

console**.**log("Marks: 95, Grade: " **+** gradestudents(95))**;**

console**.**log("Marks: 80, Grade: " **+** gradestudents(80))**;**

console**.**log("Marks: 60, Grade: " **+** gradestudents(60))**;**

console**.**log("Marks: 40, Grade: " **+** gradestudents(40))**;**

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

Ans.

Loops are programming constructs used to execute a block of code repeatedly as long as a specified condition is true. They are essential for automating repetitive tasks, iterating over collections of data, and performing operations a fixed number of times. Loops help in writing more concise and efficient code by reducing redundancy.

There are primarily three types of loops in most programming languages:-

* **for loop**: Executes a block of code a specified number of times.

Syntax: for (initialization; condition; increment/decrement) {

// code block to be executed

}

Example: for (let i = 0; i < 5; i++) {

console.log("Iteration " + i);

}

* **while loop**: Executes a block of code as long as the specified condition evaluates to true.

Syntax: while (condition) {

// code block to be executed

}

Example: let i = 0;

while (i < 5) {

console.log("Iteration " + i);

i++;

}

* **do-while loop**: Similar to a while loop, but it guarantees that the block of code is executed at least once before checking the condition.

Syntax: do {

// code block to be executed

} while (condition);

Example: let i = 0;

do {

console.log("Iteration " + i);

i++;

} while (i < 5);

Q4. Generate numbers between any 2 given numbers.

Ex: const num1 = 10

const num2 = 25;

Output: 11, 12, 13, ...., 25

Ans. function generateNumbers(num1, num2) {

// Check if num1 is less than num2

if (num1 >= num2) {

console.log("Error: num1 should be less than num2");

return;

}

// Iterate from num1 + 1 to num2 and print each number

for (let i = num1 + 1; i <= num2; i++) {

console.log(i);

}

}

// Test the function

const num1 = 10;

const num2 = 25;

console.log(`Numbers between ${num1} and ${num2}:`);

generateNumbers(num1, num2);

Q5. Use the while loop to print n9mbers from 1 to 25 in ascending and descending order.

Ans. console**.**log("Ascending Order:")**;**

let ascendingOrder **=** 1**;**

**while** (ascendingOrder **<=** 25) {

console**.**log(ascendingOrder)**;**

ascendingOrder**++;**

}

console**.**log("Descending Order:")**;**

let descendingOrder **=** 25**;**

**while** (descendingOrder **>=** 1) {

console**.**log(descendingOrder)**;**

descendingOrder**--;**

}