

Module -18 Assignment

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

Conditional statements in JavaScript are used to perform different actions based on different conditions. These conditions are evaluated as either **true** or **false**, and based on that, specific blocks of code are executed.

Types of Conditional Statements in JavaScript:-

1. if statement

The if statement executes a block of code only if a specified condition is true.

```
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer1.js > ...  
1  // if statement  
2  let age = 20;  
3      if (age >= 18)  
4      {  
5          console.log("You are eligible to vote.");  
6      }  
7
```

2. if...else statement

The if...else statement allows you to execute one block of code if the condition is true, and another block if the condition is false.

```
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer1.js > ...  
7  
8  // if...else statement  
9  let isRainy = false;  
10     if (isRainy)  
11     {  
12         console.log("Take an umbrella.");  
13     }  
14     else  
15     {  
16         console.log("No need for an umbrella.");  
17     }  
18
```

3. if...else if...else statement

This statement allows you to test multiple conditions and execute different blocks of code based on which condition is true.

```
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer1.js > ...
18
19 // if...else if...else statement
20 let score = 85;
21 if (score >= 90)
22 {
23     console.log("Grade: A");
24 }
25 else if (score >= 80)
26 {
27     console.log("Grade: B");
28 }
29 else if (score >= 70)
30 {
31     console.log("Grade: C");
32 }
33 else
34 {
35     console.log("Grade: F");
36 }
37
```

4. switch statement

The switch statement is used to execute one block of code out of many based on the value of an expression.

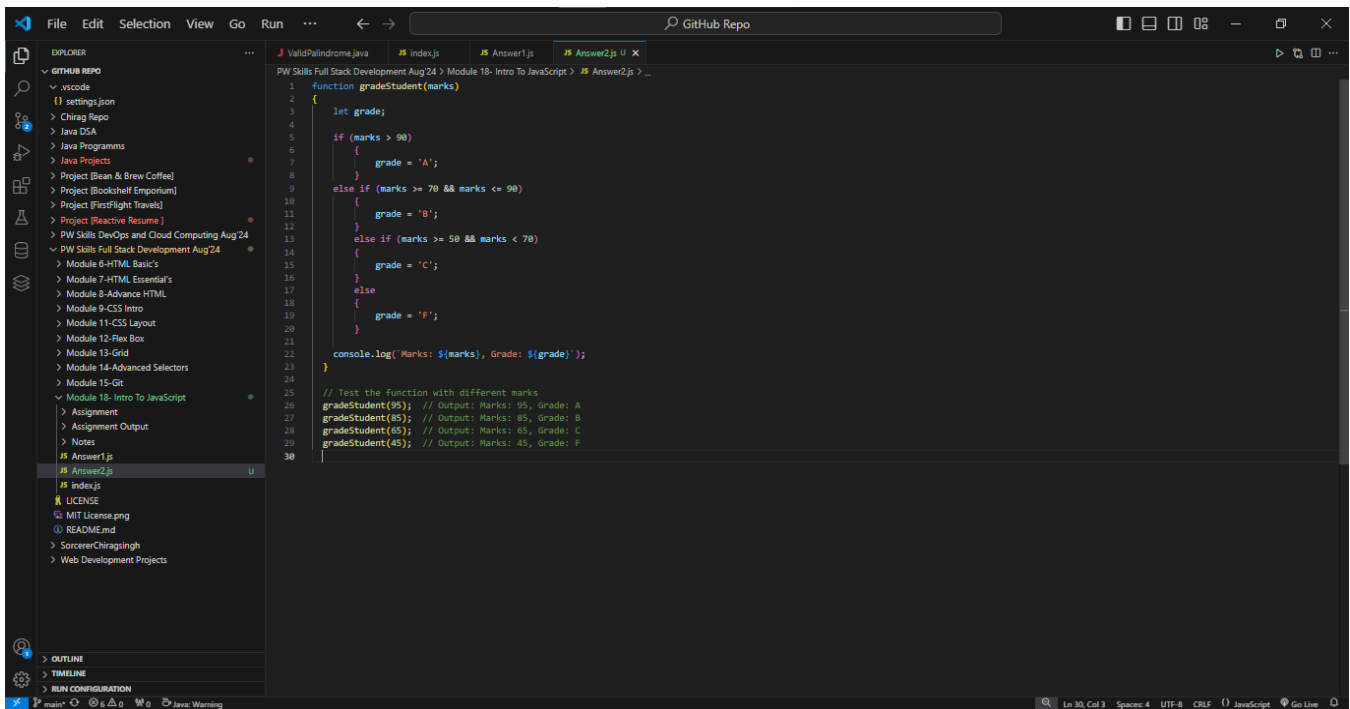
```
J ValidPalindrome.java JS index.js JS Answer1.js U X
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer1.js > ...
37
38 // switch statement
39 let day = "Tuesday";
40 switch(day)
41 {
42     case "Monday":
43         console.log("It's Monday!");
44         break;
45     case "Tuesday":
46         console.log("It's Tuesday!");
47         break;
48     default:
49         console.log("It's another day.");
50 }
51
52
```

```
[Running] node "e:\Github Repo\PW Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer1.js"
You are eligible to vote.
No need for an umbrella.
Grade: B
It's Tuesday!

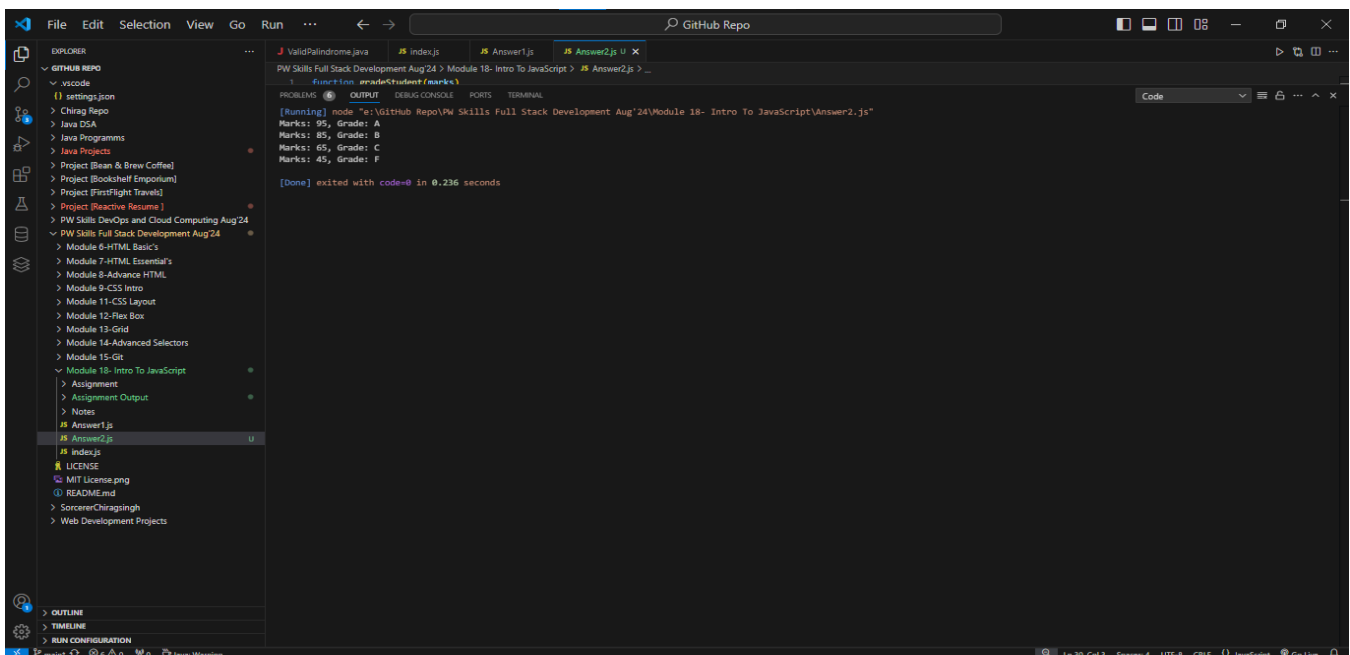
[Done] exited with code=0 in 0.263 seconds
```

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 grad
- If between 50 and 70 then a C grad
- Below 50 then an F grade



```
1 function gradeStudent(marks)
2 {
3   let grade;
4   if (marks > 90)
5   {
6     grade = 'A';
7   }
8   else if (marks >= 70 && marks <= 90)
9   {
10    grade = 'B';
11  }
12  else if (marks >= 50 && marks < 70)
13  {
14    grade = 'C';
15  }
16  else
17  {
18    grade = 'F';
19  }
20  console.log('Marks: ' + marks, 'Grade: ' + grade);
21 }
22
23 // Test the function with different marks
24 gradeStudent(95); // Output: Marks: 95, Grade: A
25 gradeStudent(85); // Output: Marks: 85, Grade: B
26 gradeStudent(65); // Output: Marks: 65, Grade: C
27 gradeStudent(45); // Output: Marks: 45, Grade: F
```



```
1 function gradeStudent(marks)
2 {
3   let grade;
4   if (marks > 90)
5   {
6     grade = 'A';
7   }
8   else if (marks >= 70 && marks <= 90)
9   {
10    grade = 'B';
11  }
12  else if (marks >= 50 && marks < 70)
13  {
14    grade = 'C';
15  }
16  else
17  {
18    grade = 'F';
19  }
20  console.log('Marks: ' + marks, 'Grade: ' + grade);
21 }
22
23 // Test the function with different marks
24 gradeStudent(95); // Output: Marks: 95, Grade: A
25 gradeStudent(85); // Output: Marks: 85, Grade: B
26 gradeStudent(65); // Output: Marks: 65, Grade: C
27 gradeStudent(45); // Output: Marks: 45, Grade: F
```

[Running] node "e:\Github Repo\PM Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer2.js"

Marks: 95, Grade: A
Marks: 85, Grade: B
Marks: 65, Grade: C
Marks: 45, Grade: F

[Done] exited with code=0 in 0.236 seconds

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

Loops in JavaScript (and programming in general) are used to repeatedly execute a block of code as long as a certain condition is met. Loops help to automate repetitive tasks, which would otherwise require writing the same code multiple times. By using loops, you can iterate over collections of data, perform repetitive actions, and save time.

Why Do We Need Loops?

- **Reduce Repetition:** Loops allow you to avoid writing the same code over and over again.
- **Work with Collections:** Loops are useful for iterating through arrays, lists, or other collections.
- **Dynamic Code Execution:** Loops make it easy to perform a task a certain number of times or until a certain condition is met.

Types of Loops in JavaScript:-

1. for loop

The for loop is used when you know how many times you want to loop beforehand. It consists of three parts: initialization, condition, and increment/decrement.

```
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer3.js > ...
1  // for loop
2  for (let i = 0; i < 5; i++)
3  {
4      console.log(i);
5  }
6
```

2. while loop

The while loop continues executing the code block as long as the specified condition remains true.

```
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer3.js > ...
7  // while loop
8  let i = 0;
9  while (i < 5)
10 {
11     console.log(i);
12     i++;
13 }
14
```

3. do...while loop

The do...while loop is similar to the while loop, but the condition is checked **after** the code block has been executed. This guarantees that the loop runs at least once.

```
PW Skills Full Stack Development Aug'24 > Module 18- Intro To JavaScript > JS Answer3.js > ...  
14  
15 // do while loop  
16 let i =0;  
17 do  
18 {  
19   console.log(i);  
20   i++;  
21 }  
22 while (i < 5);  
23
```

Output of for loop:-

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE PORTS TERMINAL  
[Running] node "e:\Github Repo\PW Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer3.js"  
0  
1  
2  
3  
4  
[Done] exited with code=0 in 0.215 seconds
```

Output of While loop: -

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE PORTS TERMINAL  
[Running] node "e:\Github Repo\PW Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer3.js"  
0  
1  
2  
3  
4  
[Done] exited with code=0 in 0.215 seconds
```

Output of Do While loop:-

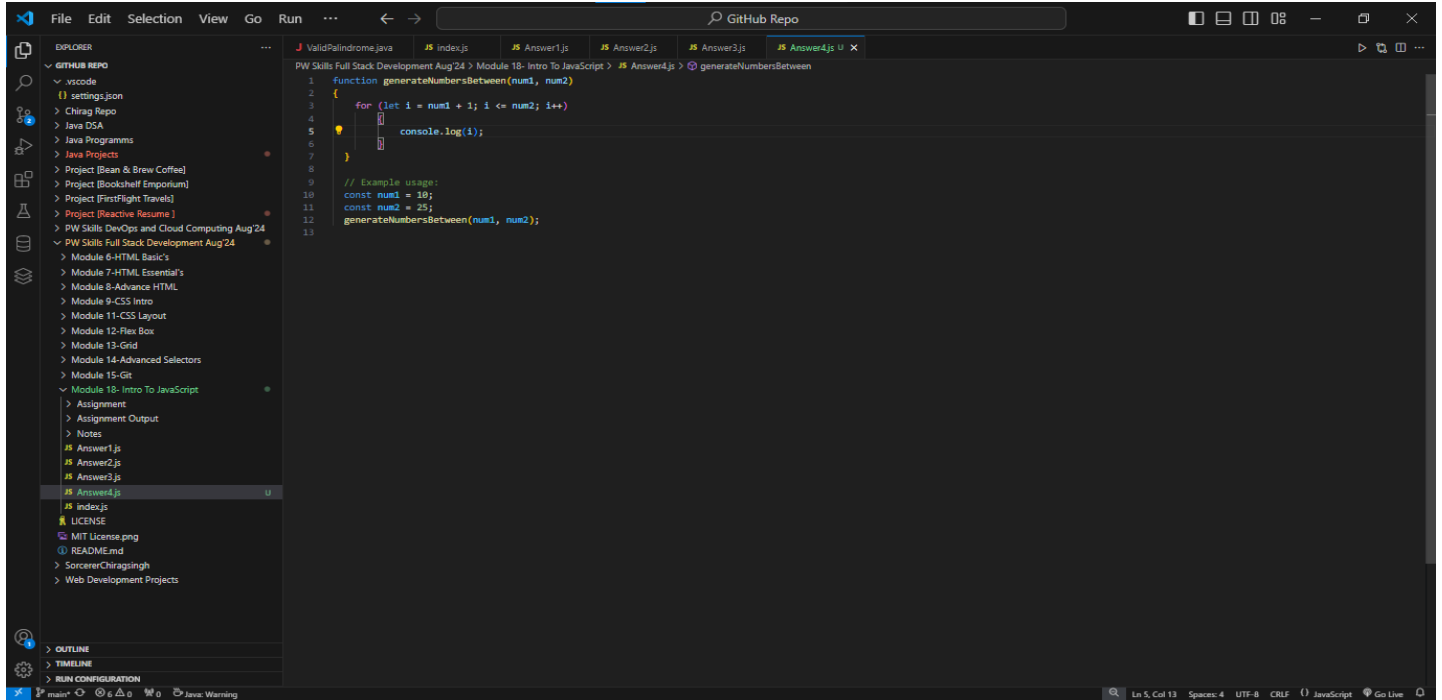
```
PROBLEMS 6 OUTPUT DEBUG CONSOLE PORTS TERMINAL  
[Running] node "e:\Github Repo\PW Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer3.js"  
0  
1  
2  
3  
4  
[Done] exited with code=0 in 0.215 seconds
```

Q4.) Generate numbers between any 2 given numbers.

Ex:

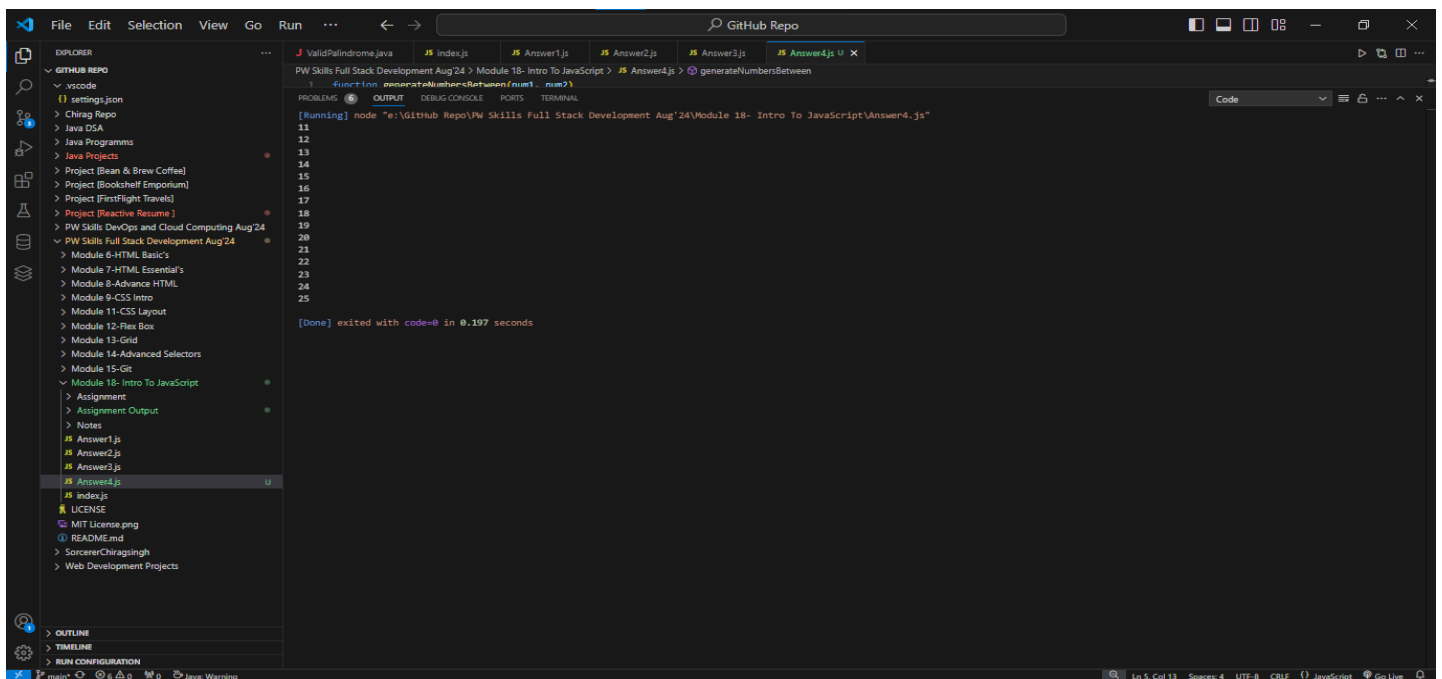
- `const num1 = 10`
- `const num2 = 25;`

Output: 11, 12, 13,, 25



The screenshot shows the VS Code editor with the file `generateNumbersBetween.js` open. The code defines a function `generateNumbersBetween(num1, num2)` that uses a `for` loop to generate numbers from `num1 + 1` to `num2` and logs them to the console. Example usage is provided with `num1 = 10` and `num2 = 25`.

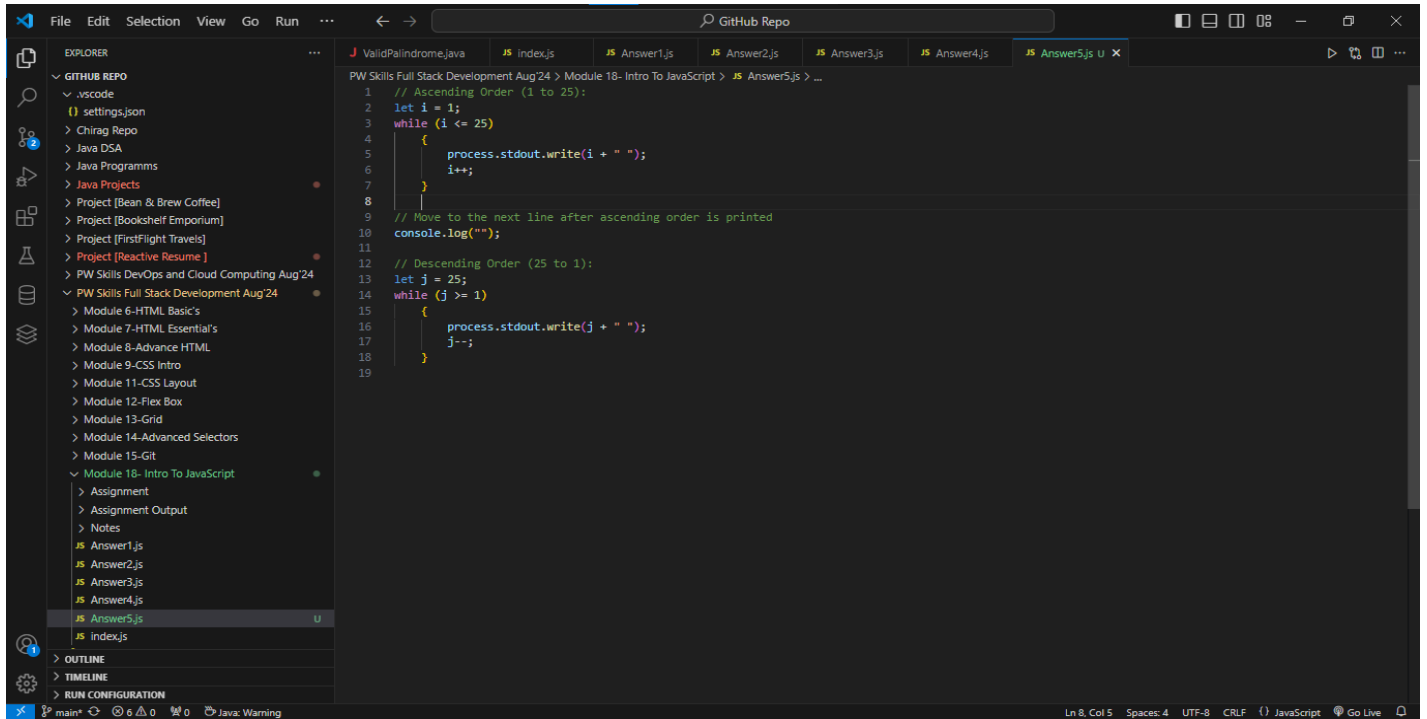
```
1 function generateNumbersBetween(num1, num2)
2 {
3     for (let i = num1 + 1; i <= num2; i++)
4     {
5         console.log(i);
6     }
7 }
8
9 // Example usage:
10 const num1 = 10;
11 const num2 = 25;
12 generateNumbersBetween(num1, num2);
13
```



The screenshot shows the VS Code editor with the terminal window open, displaying the output of the `generateNumbersBetween.js` script. The output shows numbers from 11 to 25 being logged to the console.

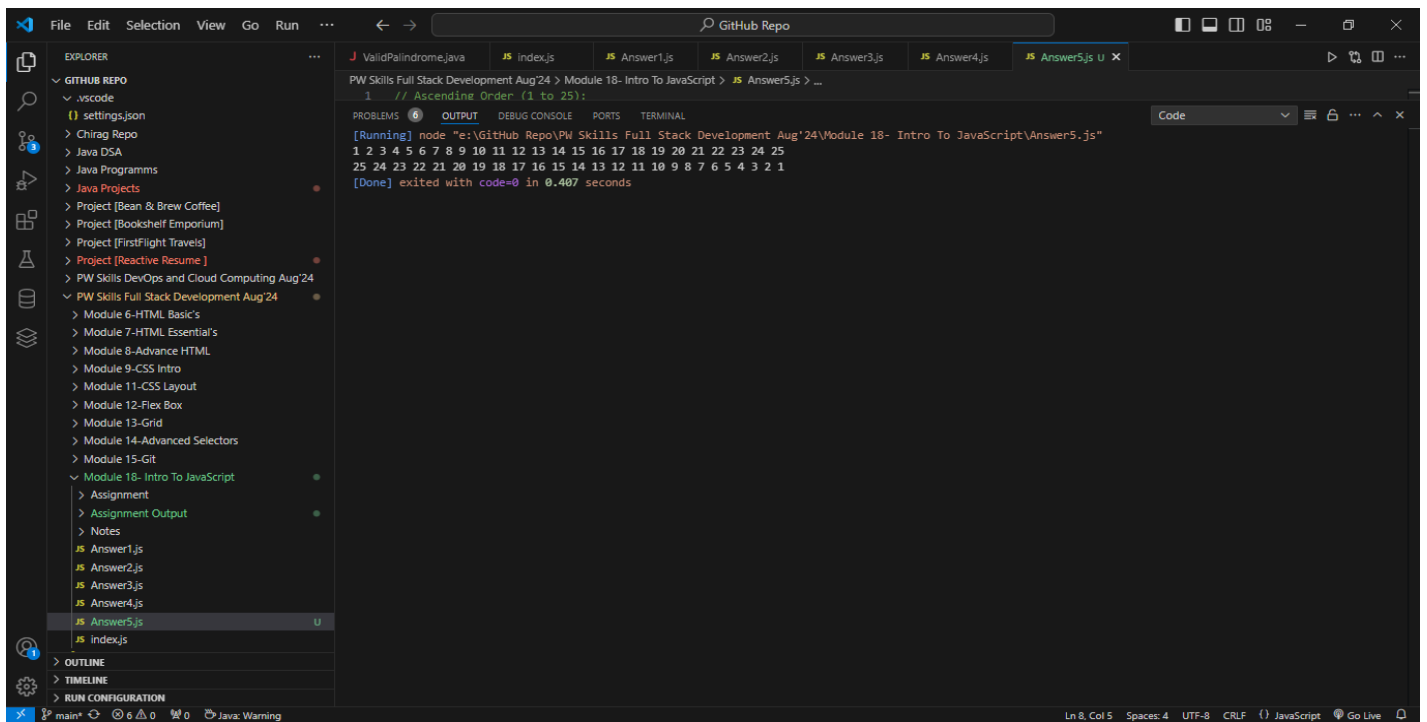
```
[Running] node "e:\Github Repo\PW Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer4.js"
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
[Done] exited with code=0 in 0.197 seconds
```

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



The screenshot shows the VS Code editor with the file `Answer5.js` open. The code is as follows:

```
1 // Ascending Order (1 to 25):
2 let i = 1;
3 while (i <= 25)
4 {
5     process.stdout.write(i + " ");
6     i++;
7 }
8
9 // Move to the next line after ascending order is printed
10 console.log("");
11
12 // Descending Order (25 to 1):
13 let j = 25;
14 while (j >= 1)
15 {
16     process.stdout.write(j + " ");
17     j--;
18 }
19
```



The screenshot shows the VS Code editor with the `OUTPUT` panel open, displaying the output of the code execution. The output is as follows:

```
[Running] node "e:\Github Repo\PW Skills Full Stack Development Aug'24\Module 18- Intro To JavaScript\Answer5.js"
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
[Done] exited with code=0 in 0.487 seconds
```