

Question 1: Output

The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project named 'CORE JAVASCRIPT 01 QUES' with a subfolder 'outputs' containing files 'question1.js' through 'question8.js'. The main editor area shows the content of 'question1.js', which is a JavaScript program to grade students based on marks. The code includes comments, variable declarations, conditional logic, and console logs. The bottom panel shows the 'TERMINAL' tab with the command 'node question1.js' executed, resulting in the output 'Marks Obtained : 90' and 'Grade : B'.

```
1  /*
2  Q1. Write a program that grades students based om their marks.
3  If greater than 90 then A Grade
4  If between 70 and 90 then a B grade
5  If between 50 and 70 then a C grade
6  Below 50 then a F grade
7  */
8
9  var marks = 90;
10 var grade;
11
12 if (marks > 90){
13     grade = "A";
14 }else if (70 < marks <= 90){
15     grade = "B";
16 }else if (50 < marks <=70){
17     grade = "C";
18 }else if (marks < 50){
19     grade = "F";
20 }
21
22 console.log("Marks Obtained :", marks);
23 console.log("Grade :", grade);
```

PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question1.js
Marks Obtained : 90
Grade : B
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques>

Question 2: Output

The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a folder named 'CORE JAVASCRIPT 01 QUES' containing an 'outputs' subfolder and several JavaScript files. The file 'question2.js' is selected and open in the editor. The code in the editor is as follows:

```
1  /*
2  Q2. Generate numbers between any 2 given numbers.
3  Ex.
4
5  Const num1 = 10;
6  Const num2 = 25;
7
8  Output : 11, 12, 13, ...., 25
9  */
10
11 var num1 = 10;
12 var num2 = 25;
13
14 console.log("num1 =", num1)
15 console.log("num2 =", num2)
16 process.stdout.write("Output: ");
17
18 while(num1 < num2){
19     num1++;
20     //console.log(num1);
21     process.stdout.write(num1 + ', ');
22 }
23
```

The bottom panel of the editor shows the 'TERMINAL' tab. It displays the command to run the script and its output:

```
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question2.js
num1 = 10
num2 = 25
Output: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques>
```

The status bar at the bottom indicates the current cursor position is at Line 15, Column 28, with 4 spaces. It also shows the file encoding as UTF-8, line endings as CRLF, and the language as JavaScript. The system tray at the very bottom shows the date and time as 10:48 on 09-01-2024.

Question 3: Output

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project named 'CORE JAVASCRIPT 01 QUES' with an 'outputs' folder containing 'question1.png' and 'question2.png'. Below these are several JavaScript files, with 'question3.js' selected. The main editor area shows the content of 'question3.js', which includes a comment, a description of the task, a variable assignment, and a ternary operator logic. The bottom panel shows the 'TERMINAL' tab with the command 'node question3.js' and its output.

File Edit Selection View Go Run ...

Core JavaScript 01 ques

EXPLORER

question1.png

question2.png

question3.js

question4..txt

question5.js

question6.js

question7.js

question8.js

question3.js

```
1 /*
2 Q3. Use a nested ternary operator to check that a number is positive, negative or zero. You have to print
3 "positive" if the number is positive and similarly for negative and zero also.
4 */
5 number = -54;
6
7 result = (number === 0) ? ("Zero") :
8 (number > 0) ? ("Positive") :
9 (number < 0) ? ("Negative") :
10 ("Please Enter a Valid number");
11
12 console.log("Nummber :", number);
13 console.log("Result :", result);
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell

```
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question3.js
Number : -54
Result : Negative
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques>
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Go Live Prettier

16°C Fog

Search

10:50 09-01-2024

Question 4: Output

The screenshot shows a code editor with a sidebar on the left containing a file explorer and an outline/timeline section. The main editor area displays the content of 'question4.txt'. The text describes the usage of the comma operator in JavaScript, providing an example of a for loop that uses it to initialize and update variables. The status bar at the bottom indicates the current cursor position (Ln 23, Col 1) and various editor settings like 'Spaces: 2', 'UTF-8', 'CRLF', 'Plain Text', 'Go Live', and 'Prettier'. The Windows taskbar is visible at the very bottom.

File Edit Selection View Go Run ...

Core JavaScript 01 ques

EXPLORER

question1.js question2.js question3.js question4.txt question5.js question6.js question7.js question8.js

CORE JAVASCRIPT 01 QUES

outputs

question1.png question2.png question3.png

question1.js question2.js question3.js question4.txt question5.js question6.js question7.js question8.js

question4.txt

1 Q4. Describe the usage of the comma operator in JavaScript and provide an example.

2 Answer:

3

4 The comma operator in JavaScript is used to combine multiple expressions into a single statement.

5 It evaluates each expression from left to right and returns the value of the last expression.

6 While it is less commonly used, it can be helpful in situations where you want to compactly express

7 several operations within a single line of code.

8

9 One common use case is in the initialization and iteration parts of a for loop.

10 Here's an example:

11

12 for (let i = 0, j = 10; i < 5; i++, j--) {

13 | console.log(i, j);

14 | }

15

16 In this example:

17

18 let i = 0, j = 10; initializes two variables (i and j) using the comma operator.

19 i++ increments the value of i in each iteration.

20 j-- decrements the value of j in each iteration.

21 console.log(i, j); prints the values of i and j in each iteration.

22

23

Ln 23, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier

16°C Fog

Search

10:52 09-01-2024

Question 5: Output

The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project named 'CORE JAVASCRIPT 01 QUES' with a folder 'outputs' and several files including 'question1.png' through 'question8.png', and 'question1.js' through 'question8.js'. The file 'question5.js' is selected and open in the editor. The code in 'question5.js' is as follows:

```
1  /*
2  Q5. You're creating a basic login system. Make a login function with two things: a username and a
3  password. Check if the username is "admin" and the password is "12345". If they're both correct, show
4  "Login Successful"; if not, show "Invalid credentials."
5  */
6
7
8  let username = "admin";
9  let password = "12345";
10
11 console.log("username :", username)
12 console.log("password :", password)
13
14 if (username == "admin" && password === "12345"){
15     console.log("Login Successful");
16 }else {
17     console.log("Invalid Credentials")
18 }
```

Below the editor, the TERMINAL panel is active, showing the command prompt output of running 'node question5.js':

```
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question5.js
username : admin
password : 12345
Login Successful
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> |
```

The status bar at the bottom indicates the current file is 'question5.js' at line 11, column 36, with 4 spaces, UTF-8 encoding, and CRLF line endings. It also shows the language is JavaScript and that Prettier is installed.

Question 6: Output

The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project named 'CORE JAVASCRIPT 01 QUES' with a subfolder 'outputs' containing question images and a list of JavaScript files. 'question6.js' is selected. The main editor area shows the code for 'question6.js', which is a JavaScript program for an e-commerce payment system. The code uses a switch statement to calculate processing fees for 'credit', 'debit', and 'paypal' methods. The bottom panel contains the 'TERMINAL' tab, showing the execution of 'node question5.js' (note the typo in the image) in a PowerShell shell. The terminal output shows a successful login for 'admin' with password '12345'. The status bar at the bottom indicates the current position is Line 27, Column 49, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray at the very bottom shows a temperature of 17°C with fog, the search bar, and various application icons.

File Edit Selection View Go Run ...

Core JavaScript 01 ques

EXPLORER

question1.png
question2.png
question3.png
question4.png
question5.png
question1.js
question2.js
question3.js
question4.txt
question5.js
question6.js
question7.js
question8.js

question6.js > ...

```
1  /*
2  Q6. You are working on an e-commerce platform. Write a JavaScript program that takes the payment
3  method ("credit", "debit", "paypal") as input and uses a switch statement to determine and print the
4  processing fee associated with each payment method. For example, "credit" may have a processing fee of
5  2%, "debit" 1.5 and "paypal" 3%.
6  */
7
8  // Choose anyone payment method ("credit", "debit", "paypal")
9  var payment_method = "credit";
10 var processing_fee;
11
12 switch(payment_method){
13     case "credit":
14         processing_fee = 0.02;
15         break;
16     case "debit":
17         processing_fee = 0.015;
18         break;
19     case "paypal":
20         processing_fee = 0.03;
21         break;
22     default:
23         console.log("Please select a valid payment method");
24 }
25
26 console.log("Payment method :", payment_method);
27 console.log("Processing fee :", processing_fee);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question5.js
username : admin
password : 12345
Login Successful
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> |

Ln 27, Col 49 Spaces: 4 UTF-8 CRLF JavaScript Go Live Prettier

17°C Fog

Search

10:59 09-01-2024

Question 7: Output

The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project named 'CORE JAVASCRIPT 01 QUES' with an 'outputs' folder containing question1.png through question6.png, and JavaScript files question1.js through question8.js. The main editor window shows 'question7.js' with the following code:

```
1  /*
2  Q7. You are building a weather application. Write a JavaScript program that takes the current temperature
3  as input and uses the conditional(ternary) operator to determine and print the weather condition. If the
4  temperature is above 30°C, the condition is "Hot"; otherwise, it is "Moderate".
5  */
6
7  var current_temperature = 31;
8
9  weather_condition = (current_temperature > 30) ? "Hot" : "Moderate";
10
11 console.log("Current temperature :", current_temperature);
12 console.log("Weather condition :", weather_condition);
13
14
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question7.js
Current temperature : 31
Weather condition : Hot
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques>
```

The status bar at the bottom indicates the file is 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and the language is 'JavaScript'. It also shows 'Go Live' and 'Prettier' icons. The Windows taskbar at the very bottom shows the date as '09-01-2024' and time as '11:01'.

Question 8: Output

The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'CORE JAVASCRIPT 01 QUES' with a subfolder 'outputs' containing files 'question1.png' through 'question7.png'. The main editor area displays 'question8.js' with the following code:

```
1  /*
2  You are creating a program to calculate the sum of numbers. Write a JavaScript program that takes a
3  positive integer as input and uses a do-while loop to calculate and print the sum of all numbers from 1 to the
4  given number.
5  */
6
7  var num = 10;    // input
8  var sum = 0;
9
10 console.log("num :", num);
11 process.stdout.write("Sum of all numbers from 1 to num = ");
12
13 do{
14     sum = sum + num;
15     num--;
16 }
17 while(num > 0);
18
19 console.log(sum);
20
21
```

Below the code editor is the TERMINAL panel, which shows the execution of the program:

```
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques> node question8.js
num : 10
Sum of all numbers from 1 to num = 55
PS C:\Users\prabh\Desktop\Core JavaScript 01 ques>
```

The status bar at the bottom indicates the file is 'Ln 12, Col 1', uses 'UTF-8' encoding, and has 'CRLF' line endings. The system tray at the very bottom shows the date and time as '09-01-2024 11:08'.