

Assignment:03

Q1.

Write a program to input two numbers and find the maximum between two numbers using the conditional/ternary operator.

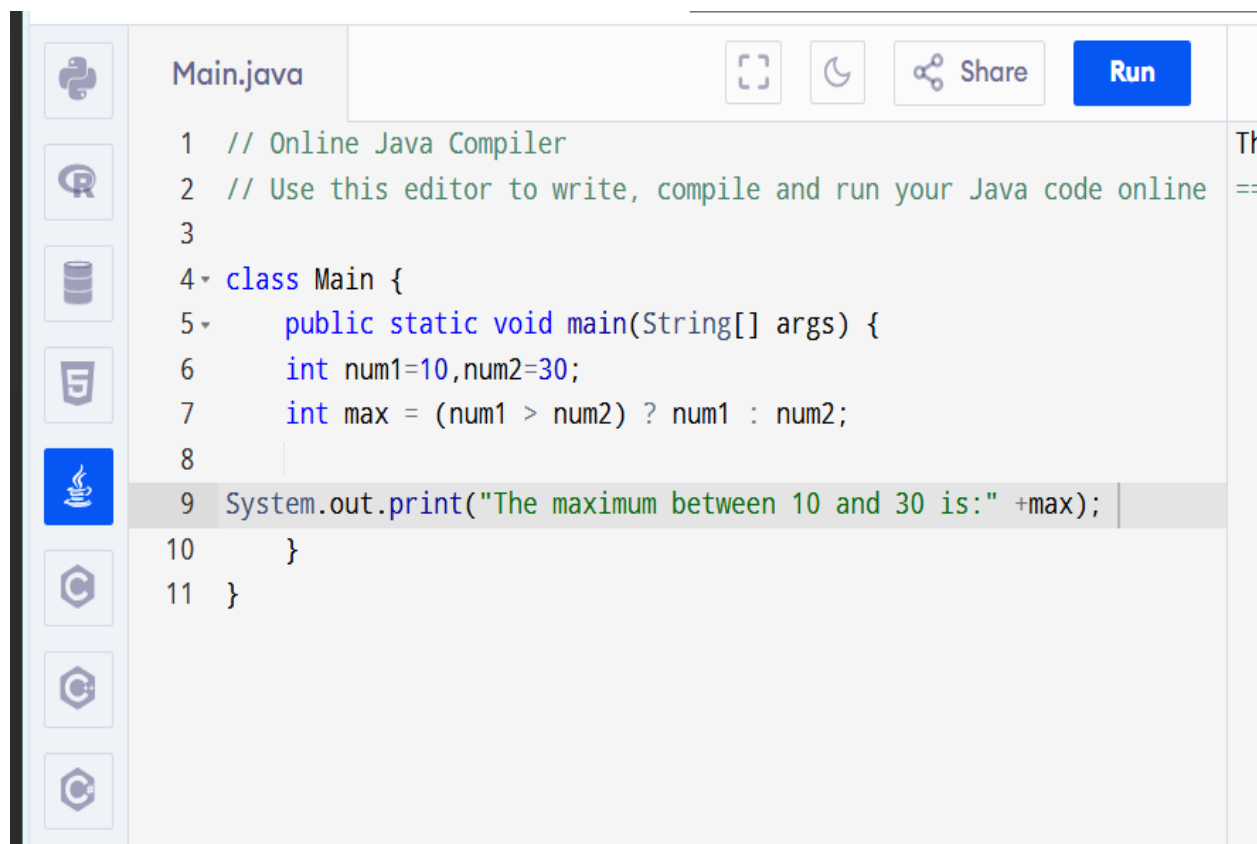
Sample Input:

num1 = 10

num2 = 30

Expected Output:

Maximum between 10 and 30 is: 30

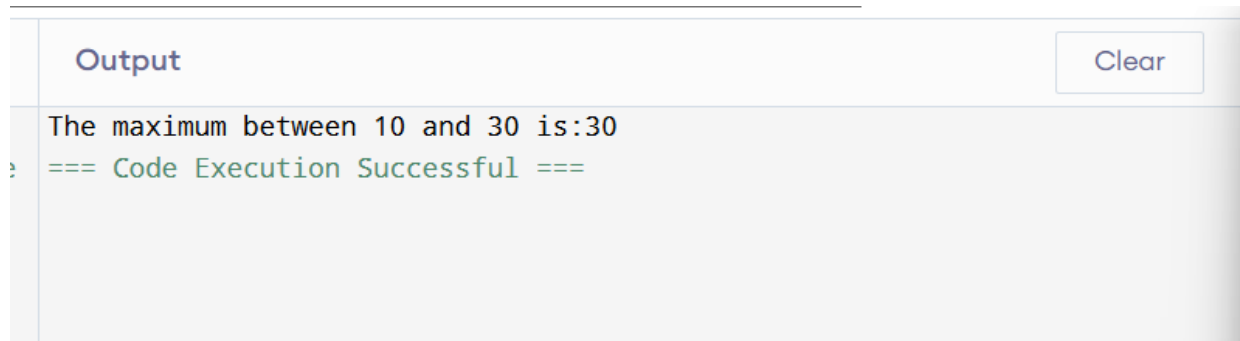


The screenshot shows an online Java compiler interface. On the left is a sidebar with icons for Python, R, SQL, JavaScript, Java (selected), and C++. The main area is titled 'Main.java' and contains the following Java code:

```
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3
4 class Main {
5     public static void main(String[] args) {
6         int num1=10,num2=30;
7         int max = (num1 > num2) ? num1 : num2;
8
9         System.out.print("The maximum between 10 and 30 is:" +max);
10    }
11 }
```

At the top right of the editor, there are icons for full screen, theme, share, and a blue 'Run' button.

OUTPUT:

A screenshot of a code execution environment's output window. The window has a title bar with the word "Output" in blue and a "Clear" button in the top right corner. The output text is displayed in a monospaced font with syntax highlighting: "The maximum between 10 and 30 is:30" in black, followed by "=== Code Execution Successful ===" in green. The background of the output area is light gray.

```
Output
```

```
The maximum between 10 and 30 is:30  
=== Code Execution Successful ===
```

Q2.

Write a program to declare two variables num and n and take an input during compilation time to check whether the nth bit of the given number is set (1) or not (0).

Logic to get nth bit of a number:

Step by step descriptive logic to get the nth bit of a number.

1. Take an input of any number and store it in some variable, say num.
2. Take an input the bit position and store it in some variable, say n.
3. To get the nth bit of num right shift num, n times. Then perform bitwise AND with 1 i.e.

```
bitStatus = (num >> n) & 1;
```

Sample Input:

Input number: num = 12

Input nth bit number: n = 2

Expected Output:

Bit 2 of 12 is set (1)



```
1 import java.util.Scanner;
2
3 class NthBitCheck {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Enter a number: ");
8         int num = sc.nextInt();
9
10        System.out.print("Enter bit position (n): ");
11        int n = sc.nextInt();
12
13        int bitStatus = (num >> n) & 1;
14
15        if(bitStatus == 1)
16            System.out.println("Bit " + n + " of " + num + " is
17                               set (1).");
18        else
19            System.out.println("Bit " + n + " of " + num + " is
20                               not set (0).");
21    }
22 }
```

Output: Enter a number: 1
Enter bit position (n): 0
Bit 0 of 1 is set (1).
=== Code Execution Successful ===

OUTPUT:



Output

Enter a number: 1
Enter bit position (n): 0
Bit 0 of 1 is set (1).
=== Code Execution Successful ===