

1.

The screenshot shows the Visual Studio Code interface. The left sidebar has icons for File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The Explorer sidebar shows 'OPEN EDITORS' with 'Welcome' and 'question1.java' open, and 'TASKK' with 'question1.java' and 'question2.java'. The main editor area has tabs for 'Welcome', 'question1.java', and 'question2.java'. The code in 'question1.java' is:

```
1 // Write a Java program using a for loop and if statement to print all numbers between 1 and 50 that are divisible by either 3 or 5, but not both.
2 public class question1 {
3     Run main | Debug main | Run | Debug
4     public static void main(String[] args) {
5         for (int i = 1; i <= 50; i++) {
6             if ((i % 3 == 0 || i % 5 == 0) && !(i % 3 == 0 && i % 5 == 0)) { // Check if divisible by 3 or 5 but not both
7                 System.out.println(i);
8             }
9         }
10    }
11 }
12 }
13 }
14 }
```

The bottom left of the interface shows sections for OUTLINE, TIMELINE, MYSQL, PROJECTS, and RUN CONFIGURATION.

Output:

```
3
5
6
9
10
12
18
20
21
24
25
27
33
35
36
39
40
42
48
50
```

2.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Taskk
- Editor:** The main editor window displays the Java code for "question2.java".
- Explorer:** The sidebar shows the project structure with files "Welcome", "question1.java", and "question2.java".
- Taskbar:** Shows indexing status: "Indexing completed." and Java environment: "Java: Ready".
- Bottom Status Bar:** Ln 16, Col 1, Spaces: 4, UTF-8, CRLF, Java, Go Live.

```
import java.util.Scanner;
// Write a Java program that takes a number from the user and prints all numbers from 1 to that number, but skips printing multiples of 4 using an if statement.
public class question2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number: ");
        int n = sc.nextInt();
        for (int i = 0; i<n; i++) {
            if (i % 4 == 0) {
                continue; // Skip multiples of 4
            }
            System.out.println(i);
        }
    }
}
```

Output:

The terminal output shows the execution of the Java program:

```
Enter a number:
16
1
2
3
5
6
7
9
10
11
13
14
15
```

3.

The screenshot shows a Java code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The left sidebar has sections for EXPLORER, OPEN EDITORS, and TASKS, with 'question3.java' selected. The main editor area contains the following Java code:

```
1 // Write a Java program using a for loop and if statements to print "Fizz" for numbers divisible by 3, "Buzz" for numbers divisible by 5, and "FizzBuzz" for numbers divisible by both.
2 public class question3 {
3     public static void main(String[] args) {
4         for (int i = 1; i <= 30; i++) {
5             if (i % 3 == 0 && i % 5 == 0) {
6                 System.out.println("FizzBuzz");
7             } else if (i % 3 == 0) {
8                 System.out.println("Fizz");
9             } else if (i % 5 == 0) {
10                 System.out.println("Buzz");
11             } else {
12                 System.out.println(i);
13             }
14         }
15     }
16 }
17
18 }
```

The status bar at the bottom shows 'Indexing completed.' and 'Java: Ready'. The bottom right corner indicates the current line and column: Ln 15, Col 10.

Output:

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
26
Fizz
28
29
FizzBuzz
```

4.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Taskk
- Explorer:** Shows open editors: Welcome, question1.java, question2.java 1, question3.java, and question4.java 1 (highlighted).
- Taskbar:** question4.java 1
- Code Editor:** Java code for question4.java:

```
1 // Write a Java program that asks for 10 integers and prints only those that are even and greater than 20 using if and for loops
2 import java.util.Scanner;
3
4 public class question4 {
5     Run main | Debug main | Run | Debug
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         for (int i = 1; i <= 10; i++) {
10             System.out.print("Enter number " + i + ": ");
11             int number = sc.nextInt();
12
13             if (number % 2 == 0 && number > 20) {
14                 System.out.println(number + " is even and greater than 20.");
15             }
16
17         }
18     }
19 }
```
- Bottom Status Bar:** Ln 11, Col 13 Spaces:4 UTF-8 CRLF { Java Go Live

Output:

```
Enter number 1: 22
22 is even and greater than 20.
Enter number 2: 24
24 is even and greater than 20.
Enter number 3: 23
Enter number 4: 28
28 is even and greater than 20.
Enter number 5: 24
24 is even and greater than 20.
Enter number 6: 15
Enter number 7: 12
Enter number 8: 14
Enter number 9: 16
Enter number 10: 22
22 is even and greater than 20.
```

5.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Taskk
- Editor:** The main editor window displays the Java code for `question5.java`.

```
// Write a Java program that prints the sum of even numbers and the product of odd numbers between 1 and 10 using a for loop and if condition.
public class question5 {
    public static void main(String[] args) {
        int sumEven = 0;
        int productOdd = 1;

        for (int i = 1; i <= 10; i++) {
            if (i % 2 == 0) {
                sumEven += i; // Add to sum if even
            } else {
                productOdd *= i; // Multiply to product if odd
            }
        }

        System.out.println("Sum of even numbers between 1 and 10: " + sumEven);
        System.out.println("Product of odd numbers between 1 and 10: " + productOdd);
    }
}
```
- Left Sidebar:** Explorer, Task, Outline, Timeline, MySQL, Projects, Run Configuration, Java Projects.
- Bottom Status Bar:** Ln 16, Col 86, Spaces:4, UTF-8, CRLF, Java, Go Live, etc.

Output:

```
Sum of even numbers between 1 and 10: 30
Product of odd numbers between 1 and 10: 945
```

6.

The screenshot shows a Java code editor in VS Code. The code is named `question6.java` and contains the following logic:

```
1 // Write a Java program using a for loop that prints numbers from 1 to 30, but:  
2 // Prints "skip" instead of 15,  
3 // Doubles any number divisible by 4 before printing,  
4 // And prints "done" at the end.  
5  
6 public class question6 {  
7     public static void main(String[] args) {  
8         for (int i = 1; i<=30; i++) {  
9             if (i == 15) {  
10                 System.out.println("skip");  
11             } else if (i % 4 == 0) {  
12                 System.out.println(i * 2);  
13             } else {  
14                 System.out.println(i);  
15             }  
16         }  
17         System.out.println("Done");  
18     }  
19 }  
20
```

The output of this code would be:

```
1  
2  
3  
8  
5  
6  
7  
16  
9  
10  
11  
24  
13  
14  
Skip  
32  
17  
18  
19  
40  
21  
22  
23  
48  
25  
26  
27  
56  
29  
30  
Done
```

Output:

```
1  
2  
3  
8  
5  
6  
7  
16  
9  
10  
11  
24  
13  
14  
Skip  
32  
17  
18  
19  
40  
21  
22  
23  
48  
25  
26  
27  
56  
29  
30  
Done
```

7.

The screenshot shows a Java code editor interface with a dark theme. The left sidebar contains icons for file operations, search, task management, MySQL, projects, run configuration, and Java projects. The main area displays a Java program named `question7.java`. The code uses a Scanner to read user input for a PIN and compares it against a correct PIN of 1234. It allows up to three attempts. If the correct PIN is entered, it prints "Access Granted". If incorrect, it prints "Incorrect PIN, try again". After three incorrect attempts, it prints "Account locked". The code editor includes tabs for other files like `Welcome`, `question1.java`, etc., and a status bar at the bottom indicating indexing completed and Java ready.

```
1 // Write a Java program that simulates an ATM PIN verification system.
2 // The program should have a correct PIN stored (for example, 1234) and allow the user up to three attempts to enter the correct PIN.
3 // If the user enters the correct PIN, print "Access Granted" and stop the program.
4 // If the user enters the wrong PIN, print "Incorrect PIN, try again".
5 // After three incorrect attempts, print "Account locked". using a for loop and if statements.
6 import java.util.Scanner;
7 public class question7 {
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        int correctPIN = 1234;
11        int attempts = 3;
12        for (int i = 1; i <= attempts; i++) {
13            System.out.println("Enter your PIN: ");
14            int enteredPIN = sc.nextInt();
15            if (enteredPIN == correctPIN) {
16                System.out.println("Access Granted");
17                break;
18            } else {
19                if (i < attempts) {
20                    System.out.println("Incorrect PIN, try again");
21                } else {
22                    System.out.println("Account locked");
23                }
24            }
25        }
26    }
27 }
28 }
```

Ln 29, Col 1 Spaces:4 UTF-8 CRLF ⚡ Java Go Live

Output:

The screenshot shows a terminal window with the path `IT\DC040327\890C\Scalat.java\Jul_19\Task_1\src\main\java\com\javatechie\atm`. The program prompts for a PIN. When "12" is entered, it prints "Incorrect PIN, try again". When "123" is entered, it prints "Incorrect PIN, try again". When "1234" is entered, it prints "Access Granted".

```
Enter your PIN:
12
Incorrect PIN, try again
Enter your PIN:
123
Incorrect PIN, try again
Enter your PIN:
1234
Access Granted
```