



**BSC. (HONS) IN ELECTRONICS AND TELECOMMUNICATIONS
ENGINEERING**

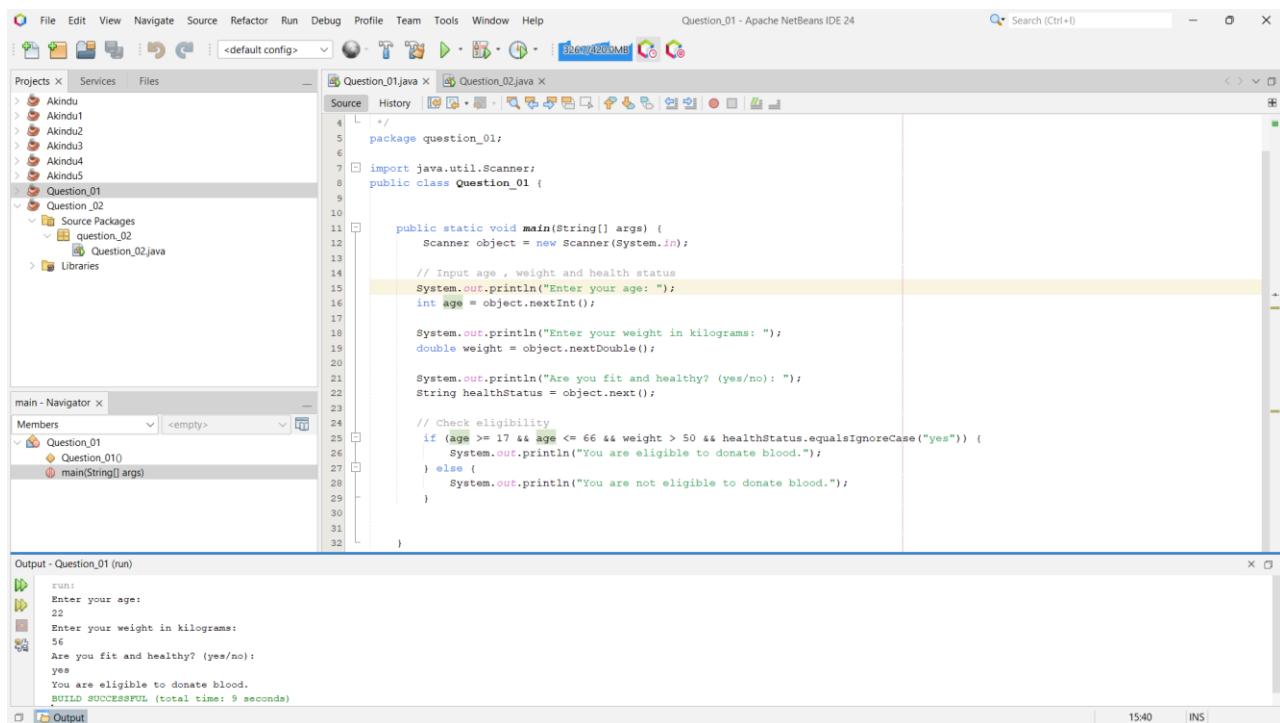
ECS2301 – Software Engineering and Project

LAB ASSIGNMENT NO. : 02

INDEX NUMBER : 23UG1- 0152_Akindu Randira

7th JANUARY 2025

1. You should be able to give blood if you are fit and healthy, weigh over 50kg and are between 17 and 66 years. Write a simple program to check the eligibility of blood donation.



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "Question_01 - Apache NetBeans IDE 24". The left sidebar has a Projects tab with several projects listed, and a Files tab showing files like Question_01.java and Question_02.java. The main workspace displays the source code for Question_01.java:

```

4  /*
5   * package question_01;
6
7   import java.util.Scanner;
8   public class Question_01 {
9
10
11    public static void main(String[] args) {
12        Scanner object = new Scanner(System.in);
13
14        // Input age , weight and health status
15        System.out.println("Enter your age: ");
16        int age = object.nextInt();
17
18        System.out.println("Enter your weight in kilograms: ");
19        double weight = object.nextDouble();
20
21        System.out.println("Are you fit and healthy? (yes/no): ");
22        String healthStatus = object.next();
23
24        // Check eligibility
25        if (age >= 17 && age <= 66 && weight > 50 && healthStatus.equalsIgnoreCase("yes")) {
26            System.out.println("You are eligible to donate blood.");
27        } else {
28            System.out.println("You are not eligible to donate blood.");
29        }
30
31    }
32

```

The Output window at the bottom shows the run results:

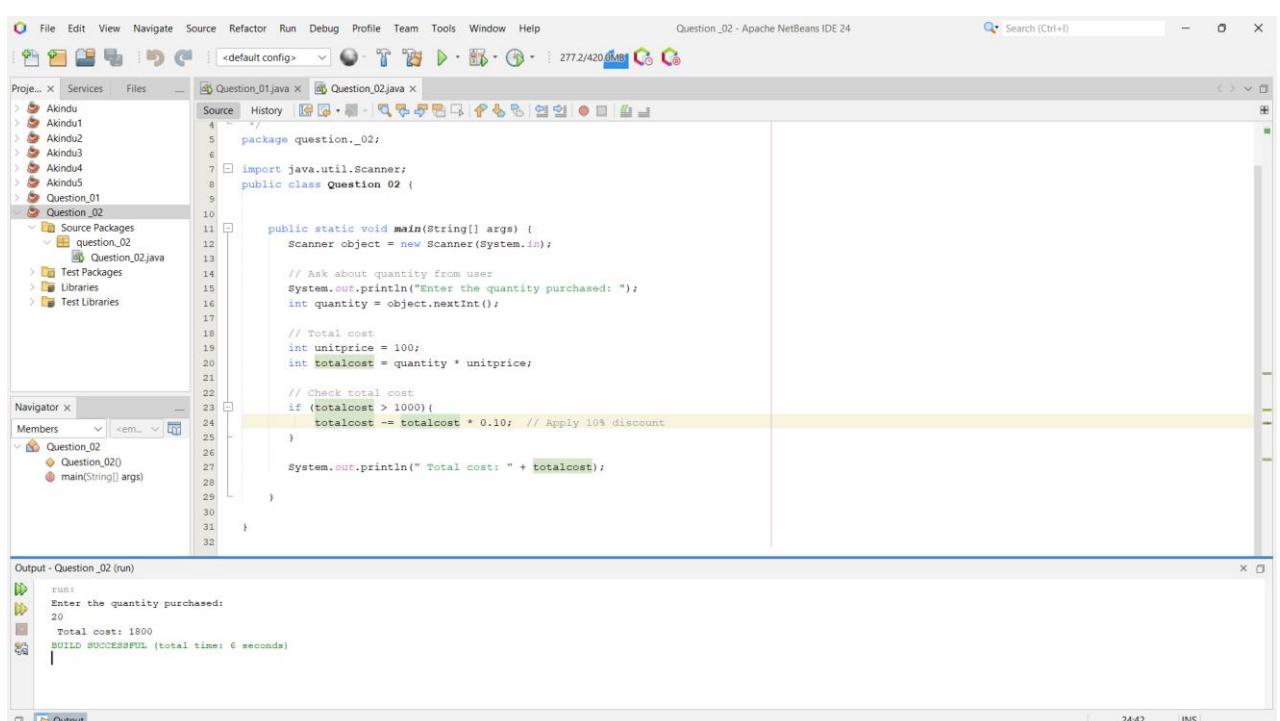
```

run:
Enter your age:
22
Enter your weight in kilograms:
56
Are you fit and healthy? (yes/no):
yes
You are eligible to donate blood.
BUILD SUCCESSFUL (total time: 9 seconds)

```

2. A shop will give a discount of 10% if the cost of the quantity purchased is more than 1000.

- Ask user for quantity
- Suppose, one unit will cost 100.
- Judge and print total cost for the user.



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "Question_02 - Apache NetBeans IDE 24". The left sidebar has a Projects tab with several projects listed, and a Files tab showing files like Question_01.java and Question_02.java. The main workspace displays the source code for Question_02.java:

```

4  /*
5   * package question_02;
6
7   import java.util.Scanner;
8   public class Question_02 {
9
10
11    public static void main(String[] args) {
12        Scanner object = new Scanner(System.in);
13
14        // Ask about quantity from user
15        System.out.println("Enter the quantity purchased: ");
16        int quantity = object.nextInt();
17
18        // Total cost
19        int unitprice = 100;
20        int totalcost = quantity * unitprice;
21
22        // Check total cost
23        if (totalcost > 1000){
24            totalcost -= totalcost * 0.10; // Apply 10% discount
25        }
26
27        System.out.println(" Total cost: " + totalcost);
28
29    }
30
31
32

```

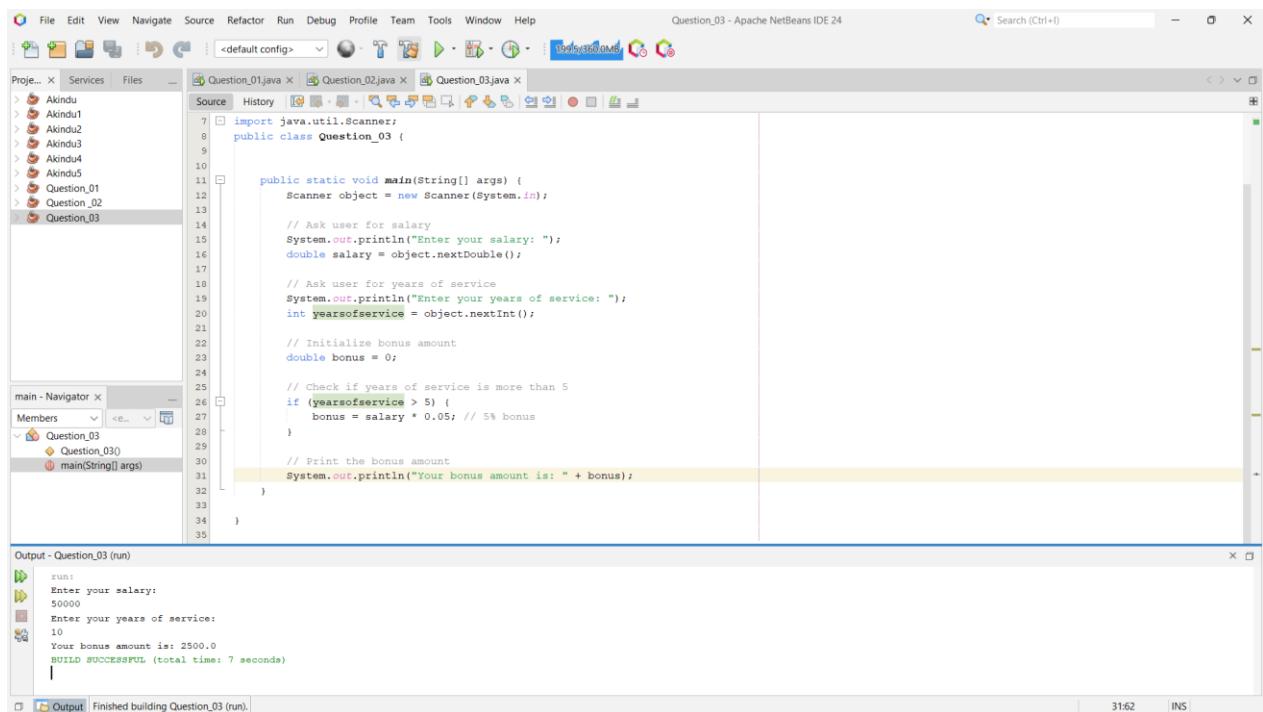
The Output window at the bottom shows the run results:

```

run:
Enter the quantity purchased:
20
Total cost: 1800
BUILD SUCCESSFUL (total time: 6 seconds)

```

3. A company decided to give a bonus of 5% to an employee if his/her year of service is more than 5 years. Ask users for their salary and year of service and print the net bonus amount.



```

import java.util.Scanner;
public class Question_03 {
    public static void main(String[] args) {
        Scanner object = new Scanner(System.in);

        // Ask user for salary
        System.out.println("Enter your salary: ");
        double salary = object.nextDouble();

        // Ask user for years of service
        System.out.println("Enter your years of service: ");
        int yearsofservice = object.nextInt();

        // Initialize bonus amount
        double bonus = 0;

        // Check if years of service is more than 5
        if (yearsofservice > 5) {
            bonus = salary * 0.05; // 5% bonus
        }

        // Print the bonus amount
        System.out.println("Your bonus amount is: " + bonus);
    }
}

```

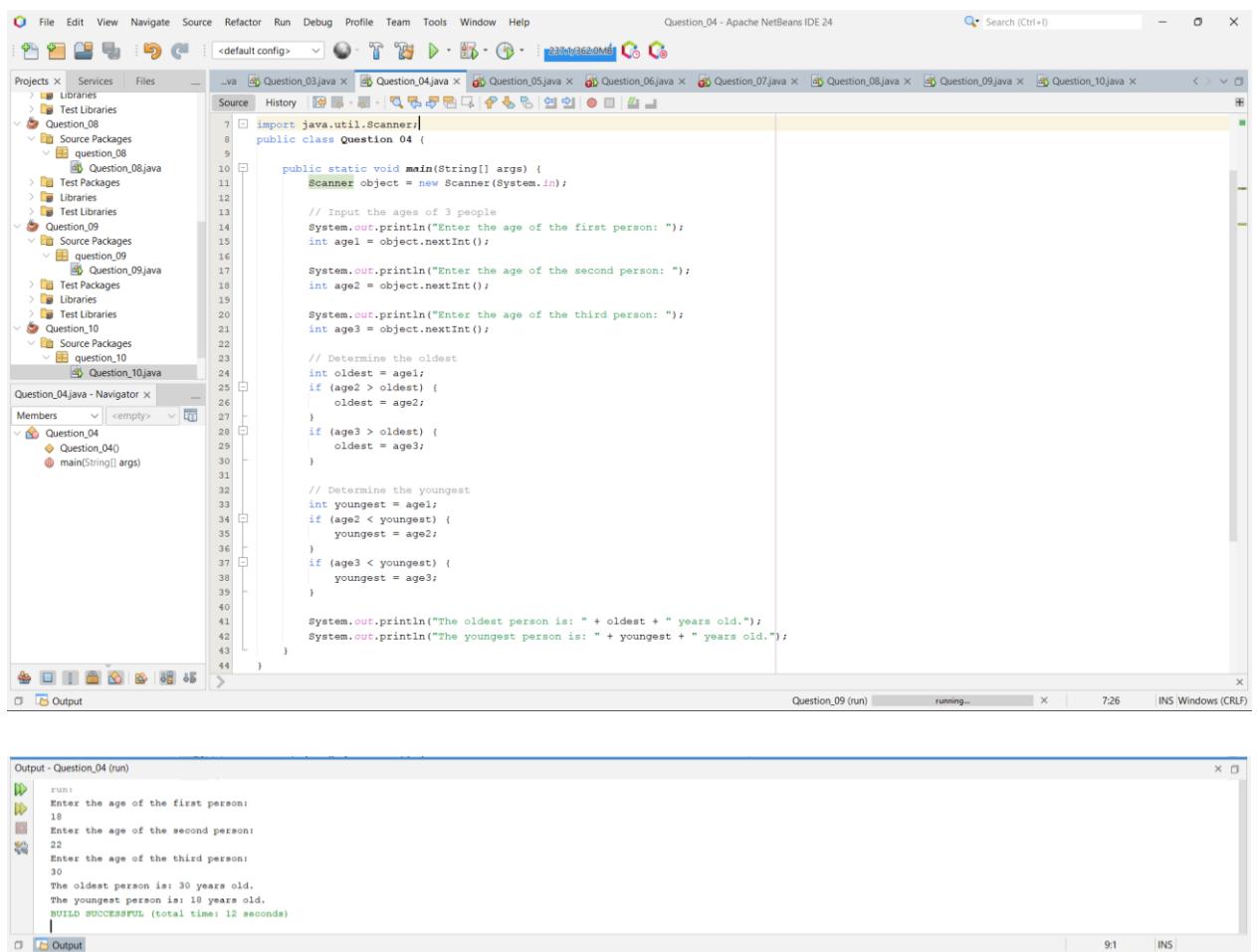
Output - Question_03 (run)

```

run:
Enter your salary:
50000
Enter your years of service:
10
Your bonus amount is: 2500.0
BUILD SUCCESSFUL (total time: 7 seconds)

```

4. Take input of the age of 3 people by user and determine oldest and youngest among them.



```

import java.util.Scanner;
public class Question_04 {
    public static void main(String[] args) {
        Scanner object = new Scanner(System.in);

        // Input the ages of 3 people
        System.out.println("Enter the age of the first person: ");
        int age1 = object.nextInt();

        System.out.println("Enter the age of the second person: ");
        int age2 = object.nextInt();

        System.out.println("Enter the age of the third person: ");
        int age3 = object.nextInt();

        // Determine the oldest
        int oldest = age1;
        if (age2 > oldest) {
            oldest = age2;
        }
        if (age3 > oldest) {
            oldest = age3;
        }

        // Determine the youngest
        int youngest = age1;
        if (age2 < youngest) {
            youngest = age2;
        }
        if (age3 < youngest) {
            youngest = age3;
        }

        System.out.println("The oldest person is: " + oldest + " years old.");
        System.out.println("The youngest person is: " + youngest + " years old.");
    }
}

```

Output - Question_04 (run)

```

run:
Enter the age of the first person:
18
Enter the age of the second person:
22
Enter the age of the third person:
30
The oldest person is: 30 years old.
The youngest person is: 18 years old.
BUILD SUCCESSFUL (total time: 12 seconds)

```

5. Write a program to check whether an entered character is lowercase (a to z) or uppercase (A to Z).

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package question_05;

import java.util.Scanner;
public class Question_05 {

    public static void main(String[] args) {
        Scanner object = new Scanner(System.in);

        // Input character from user
        System.out.println("Enter a character: ");
        char ch = object.next().charAt(0);

        // Check if the character is uppercase or lowercase
        if (ch >= 'a' && ch <= 'z') {
            System.out.println("The character '" + ch + "' is lowercase.");
        } else if (ch >= 'A' && ch <= 'Z') {
            System.out.println("The character '" + ch + "' is uppercase.");
        } else {
            System.out.println("The character '" + ch + "' is not an alphabet letter.");
        }
    }
}

```

6. The two roots of a quadratic equation $ax^2 + bx + c = 0$ can be obtained using the following

$$\text{formula: } r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$b^2 - 4ac$ is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots. Write a program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display "The equation has no real roots". Note that you can use `Math.pow(x, 0.5)` to compute $2x$. Here are some sample runs.

Enter a, b, c: 1.0 3 1

The equation has two roots -0.381966 and -2.61803

Enter a, b, c: 1 2.0 1

The equation has one root -1

Enter a, b, c: 1 2 3 The equation has no real roots

```

package question_06;
import java.util.Scanner;
public class Question_06 {
    public static void main(String[] args) {
        Scanner object = new Scanner(System.in);

        // Prompt user for inputs
        System.out.println("Enter coefficients a, b, c:");
        double a = object.nextDouble();
        double b = object.nextDouble();
        double c = object.nextDouble();

        // Calculate the discriminant
        double discriminant = Math.pow(b, 2) - 4 * a * c;

        // Check the discriminant and compute roots
        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The equation has two roots: " + root1 + " and " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("The equation has one root: " + root);
        } else {
            System.out.println("The equation has no real roots");
        }
    }
}

```

Output - Question_06 (run)

```

run:
Enter coefficients a, b, c:
1 2 1
The equation has one root: -1.0
BUILD SUCCESSFUL (total time: 11 seconds)

```

7. A student will not be allowed to sit an exam if his/her attendance is less than 75%. Take following input from user

a. Number of classes held

b. Number of classes attended. And print

c. percentage of class attended d. Is the student allowed to sit in the exam or not.

```

import java.util.Scanner;
public class Question_07 {
    public static void main(String[] args) {
        Scanner object = new Scanner(System.in);

        // Take input for classes held and attended
        System.out.println("Enter the number of classes held: ");
        int classesheld = object.nextInt();

        System.out.println("Enter the number of classes attended: ");
        int classesattended = object.nextInt();

        // Calculate attendance percentage
        double attendancepercentage = ((double) classesattended / classesheld) * 100;

        // Display attendance percentage
        System.out.println("Percentage of classes attended: " + attendancepercentage + "%");

        // Check if student is allowed to sit in the exam
        if (attendancepercentage >= 75) {
            System.out.println("The student is allowed to sit in the exam.");
        } else {
            System.out.println("The student is not allowed to sit in the exam.");
        }
    }
}

```

Output - Question_07 (run)

```

run:
Enter the number of classes held:
20
Enter the number of classes attended:
18
Percentage of classes attended: 90.0%
The student is allowed to sit in the exam.
BUILD SUCCESSFUL (total time: 7 seconds)

```

8. Write a program that randomly generates an integer between 1 and 12 and displays the English month name January, February, ..., December for the number 1, 2, ..., 12, accordingly.

The screenshot shows the NetBeans IDE interface with multiple Java files listed in the top tab bar. The main code editor displays the following Java code:

```
7 import java.util.Random;
8 public class Question_08 {
9
10
11    public static void main(String[] args) {
12        Random object = new Random();
13
14        // Generate a random integer between 1 and 12
15        int monthNumber = object.nextInt(12) + 1;
16
17        // Determine the corresponding month name
18        String monthName;
19        switch (monthNumber) {
20            case 1:
21                monthName = "January";
22                break;
23            case 2:
24                monthName = "February";
25                break;
26            case 3:
27                monthName = "March";
28                break;
29            case 4:
30                monthName = "April";
31                break;
32            case 5:
33                monthName = "May";
34                break;
35            case 6:
36                monthName = "June";
37                break;
38            case 7:
39                monthName = "July";
40                break;
41            case 8:
42                monthName = "August";
43                break;
44            case 9:
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66 }
```

The code uses a switch statement to map integers from 1 to 12 to their corresponding month names. The output window shows the program running and printing the result.

The screenshot shows the NetBeans IDE interface with multiple Java files listed in the top tab bar. The main code editor displays the same Java code as the previous screenshot, but the 'default' case in the switch statement is highlighted with a yellow background.

```
41     case 8:
42         monthName = "August";
43         break;
44     case 9:
45         monthName = "September";
46         break;
47     case 10:
48         monthName = "October";
49         break;
50     case 11:
51         monthName = "November";
52         break;
53     case 12:
54         monthName = "December";
55         break;
56     default:
57         monthName = "Invalid Month";
58     }
59
60     // Display the result
61     System.out.println("The randomly generated month number is: " + monthNumber);
62     System.out.println("The corresponding month is: " + monthName);
63 }
```

The code is identical to the one in the first screenshot, except for the visual highlighting of the 'default' case. The output window shows the program running and printing the result.

9. An ISBN-10 (International Standard Book Number) consists of 10 digits:
d₁d₂d₃d₄d₅d₆d₇d₈d₉d₁₀. The last digit, d₁₀, is a checksum, which is calculated from the other nine digits using the following formula:

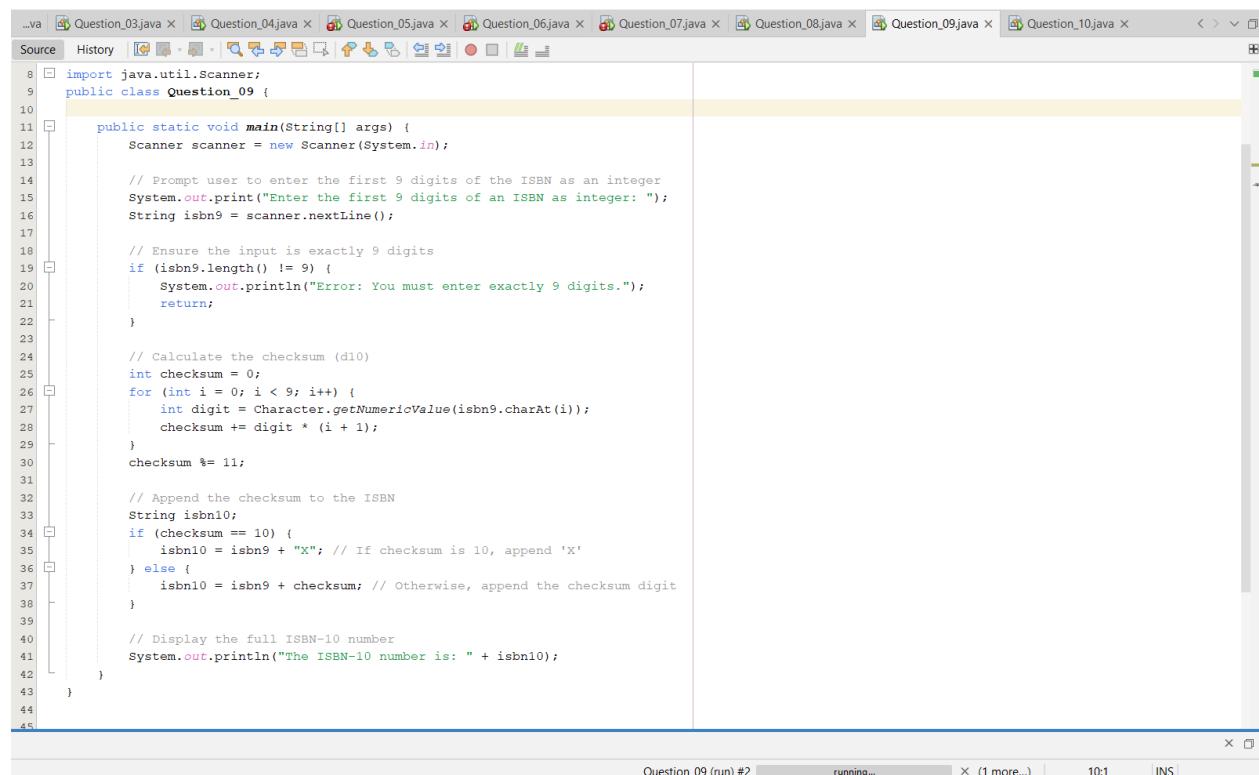
$$(d_1 * 1 + d_2 * 2 + d_3 * 3 + d_4 * 4 + d_5 * 5 + d_6 * 6 + d_7 * 7 + d_8 * 8 + d_9 * 9) \% 11$$

If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention.

Write a program that prompts the user to enter the first 9 digits and displays the 10-digit ISBN (including leading zeros). Your program should read the input as an integer. Here are sample runs:

Enter the first 9 digits of an ISBN as integer: 013601267 The ISBN-10 number is 0136012671

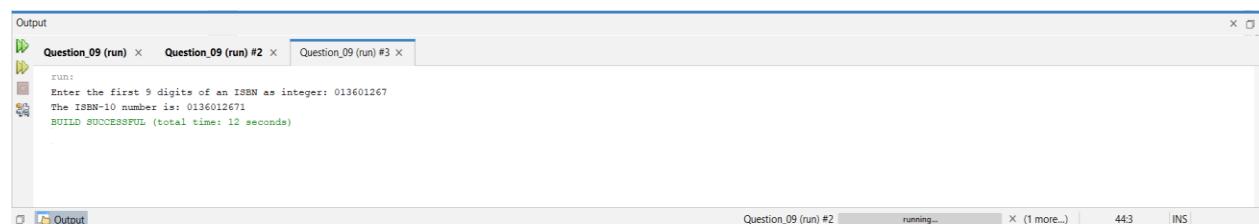
Enter the first 9 digits of an ISBN as integer: 013031997 The ISBN-10 number is 013031997X



The screenshot shows an IDE interface with multiple tabs at the top labeled Question_03.java, Question_04.java, Question_05.java, Question_06.java, Question_07.java, Question_08.java, Question_09.java, and Question_10.java. The main editor window contains Java code for a class named Question_09. The code prompts the user for the first 9 digits of an ISBN, calculates the checksum (d10), and prints the full 10-digit ISBN. The code uses Scanner for input and System.out for output.

```
8 import java.util.Scanner;
9 public class Question_09 {
10
11     public static void main(String[] args) {
12         Scanner scanner = new Scanner(System.in);
13
14         // Prompt user to enter the first 9 digits of the ISBN as an integer
15         System.out.print("Enter the first 9 digits of an ISBN as integer: ");
16         String isbn9 = scanner.nextLine();
17
18         // Ensure the input is exactly 9 digits
19         if (isbn9.length() != 9) {
20             System.out.println("Error: You must enter exactly 9 digits.");
21             return;
22         }
23
24         // Calculate the checksum (d10)
25         int checksum = 0;
26         for (int i = 0; i < 9; i++) {
27             int digit = Character.getNumericValue(isbn9.charAt(i));
28             checksum += digit * (i + 1);
29         }
30         checksum %= 11;
31
32         // Append the checksum to the ISBN
33         String isbn10;
34         if (checksum == 10) {
35             isbn10 = isbn9 + "x"; // If checksum is 10, append 'x'
36         } else {
37             isbn10 = isbn9 + checksum; // Otherwise, append the checksum digit
38         }
39
40         // Display the full ISBN-10 number
41         System.out.println("The ISBN-10 number is: " + isbn10);
42     }
43 }
44
45
```

Question_09 (run) #2 running... × (1 more...) | 10:1 | INS



The screenshot shows the IDE's Output window with three tabs: Question_09 (run), Question_09 (run) #2, and Question_09 (run) #3. The Question_09 (run) tab is active and displays the execution results of three runs. Run #2 shows the user input '013601267' and the output 'The ISBN-10 number is: 0136012671'. Run #3 shows the user input '013031997' and the output 'The ISBN-10 number is: 013031997X'. The Output window also shows a BUILD SUCCESSFUL message with a total time of 12 seconds.

Output

Question_09 (run) × Question_09 (run) #2 × Question_09 (run) #3 ×

run:
Enter the first 9 digits of an ISBN as integer: 013601267
The ISBN-10 number is: 0136012671
BUILD SUCCESSFUL (total time: 12 seconds)

Question_09 (run) #2 running... × (1 more...) | 443 | INS

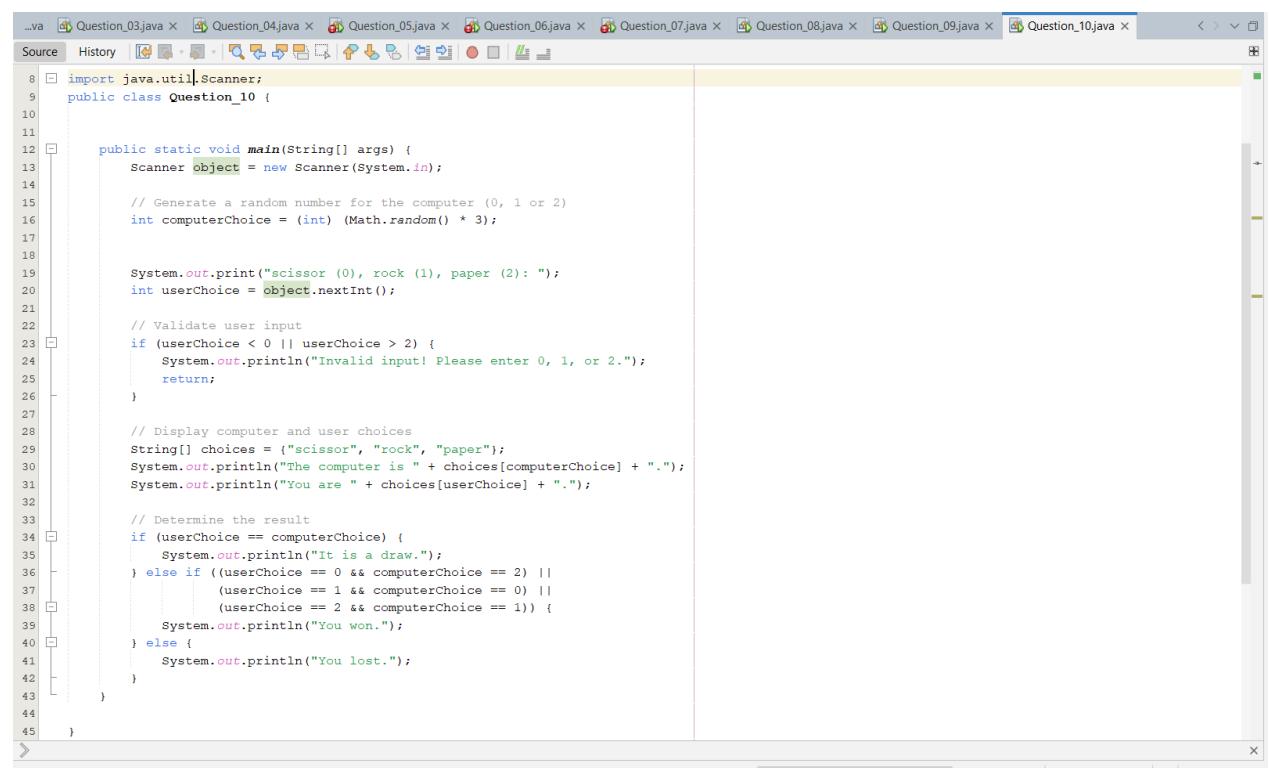
10. (Game: scissor, rock, paper) Write a program that plays the popular scissor-rockpaper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates a number 0, 1, or 2 representing scissor, rock, and paper. The program prompts the user to enter a number 0, 1, or 2 and displays a message indicating whether the user or the computer wins, loses, or draws. Here are sample runs:

scissor (0), rock (1), paper (2): 1

The computer is scissor. You are rock. You won

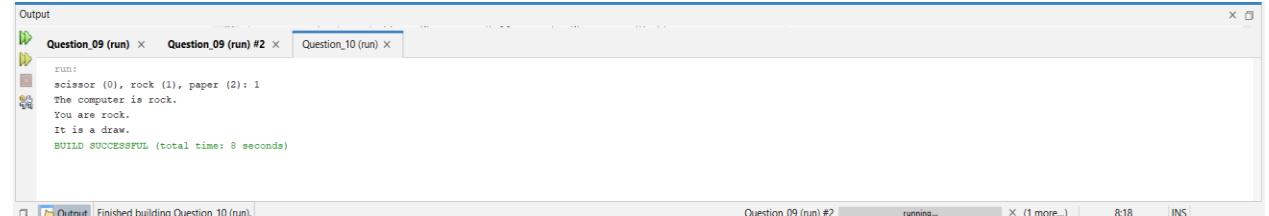
scissor (0), rock (1), paper (2): 2

The computer is paper. You are paper too. It is a draw



The screenshot shows an IDE interface with multiple tabs at the top labeled Question_03.java, Question_04.java, Question_05.java, Question_06.java, Question_07.java, Question_08.java, Question_09.java, and Question_10.java. The Question_10.java tab is active. Below the tabs is a toolbar with icons for file operations like new, open, save, and run. The main area contains the Java code for the game. The code imports java.util.Scanner and defines a public class Question_10. It includes a main method that uses Scanner to get user input, generates a random computer choice (0, 1, or 2), prints the choices, validates user input, and determines the result based on the rules of the game. The output window at the bottom shows the execution of the program with sample inputs and outputs.

```
8 import java.util.Scanner;
9 public class Question_10 {
10
11
12     public static void main(String[] args) {
13         Scanner object = new Scanner(System.in);
14
15         // Generate a random number for the computer (0, 1 or 2)
16         int computerChoice = (int) (Math.random() * 3);
17
18
19         System.out.print("scissor (0), rock (1), paper (2): ");
20         int userChoice = object.nextInt();
21
22         // Validate user input
23         if (userChoice < 0 || userChoice > 2) {
24             System.out.println("Invalid input! Please enter 0, 1, or 2.");
25             return;
26         }
27
28
29         // Display computer and user choices
30         String[] choices = {"scissor", "rock", "paper"};
31         System.out.println("The computer is " + choices[computerChoice] + ".");
32         System.out.println("You are " + choices[userChoice] + ".");
33
34         // Determine the result
35         if (userChoice == computerChoice) {
36             System.out.println("It is a draw.");
37         } else if ((userChoice == 0 && computerChoice == 2) ||
38                    (userChoice == 1 && computerChoice == 0) ||
39                    (userChoice == 2 && computerChoice == 1)) {
40             System.out.println("You won.");
41         } else {
42             System.out.println("You lost.");
43         }
44     }
45 }
```



The screenshot shows the Output window of the IDE. It lists three runs: Question_09 (run) #2, Question_09 (run) #2, and Question_10 (run). The Question_10 (run) output shows the program's execution. It starts with a run command, followed by a user input of '1' (scissor), the computer's choice ('rock'), and the resulting message ('You are rock. It is a draw.'). The output concludes with a 'BUILD SUCCESSFUL' message. The status bar at the bottom indicates the build was successful with a total time of 8 seconds.

```
run:
scissor (0), rock (1), paper (2): 1
The computer is rock.
You are rock.
It is a draw.
BUILD SUCCESSFUL (total time: 8 seconds)
```