

Question2.java

```
1 import java.io.*;
2 import java.util.*;
3 import java.lang.Math.*;
4
5 /**
6  * Matrix class creates two methods to perform matrix addition and
7  * multiplication
8  *
9  */
10 class Matrix {
11
12     String header = "Matrix Operations";
13
14     Random random = new Random();
15
16     /*
17      * matrixAddition takes the size of row and column of the matrix as
18      arguments.
19      * It initializes two 2D arrays and fills them with randomly generated
20      elements,
21      * using which it performs addition and returns the sum matrix.
22      */
23     int[][] matrixAddition(int m, int n) {
24         Matrix M = new Matrix();
25
26         int[][] A = new int[m][n];
27         int[][] B = new int[m][n];
28         int[][] Sum = new int[m][n];
29
30         System.out.println("\nElements of Matrix A: ");
31         for (int i = 0; i < m; i++) {
32             for (int j = 0; j < n; j++) {
33                 A[i][j] = random.nextInt(100);
34             }
35         }
36         M.showMatrix(A, m, n);
37
38         System.out.println("\nElements of Matrix B: ");
39         for (int i = 0; i < m; i++) {
40             for (int j = 0; j < n; j++) {
41                 B[i][j] = random.nextInt(100);
42             }
43         }
44     }
45 }
```

Question2.java

```
41     }
42     M.showMatrix(B, m, n);
43
44     System.out.println("\nSum of matrix A and B: ");
45     for (int i = 0; i < m; i++) {
46         for (int j = 0; j < n; j++) {
47             Sum[i][j] = A[i][j] + B[i][j];
48         }
49     }
50     M.showMatrix(Sum, m, n);
51     return Sum;
52 }
53
54  /*
55   * matrixAddition takes the size of row and column of the matrix as
   arguments.
56   * It initializes two 2D arrays and fills them with randomly generated
   elements,
57   * using which it performs multiplication and returns the product
   matrix.
58   */
59   int[][] matrixMultiplication(int m, int n) {
60       Matrix M = new Matrix();
61
62       int[][] A = new int[m][n];
63       int[][] B = new int[m][n];
64       int[][] Product = new int[m][n];
65
66       System.out.println("\nElements of Matrix A: ");
67       for (int i = 0; i < m; i++) {
68           for (int j = 0; j < n; j++) {
69               A[i][j] = random.nextInt(100);
70           }
71       }
72       M.showMatrix(A, m, n);
73
74       System.out.println("\nElements of Matrix B: ");
75       for (int i = 0; i < m; i++) {
76           for (int j = 0; j < n; j++) {
77               B[i][j] = random.nextInt(100);
78           }
79       }
```

Question2.java

```
80         M.showMatrix(B, m, n);
81
82         System.out.println("\nProduct of matrix A and B: ");
83         for (int i = 0; i < m; i++) {
84             for (int j = 0; j < n; j++) {
85                 Product[i][j] = A[i][j] * B[i][j];
86             }
87         }
88         M.showMatrix(Product, m, n);
89         return Product;
90     }
91
92     /*
93     * showMatrix takes integer array, row size and column size as
    parameters and
94     * outputs the matrix to the console
95     */
96     private void showMatrix(int[][] arr, int m, int n) {
97         for (int i = 0; i < m; i++) {
98             for (int j = 0; j < n; j++) {
99                 System.out.print(arr[i][j] + "\t");
100             }
101             System.out.println();
102         }
103     }
104 }
105
106 /**
107  * public class includes the main method, main takes the size of row and
    column
108  * from user
109  */
110 public class Question2 {
111
112     private static Scanner in = new Scanner(System.in);
113
114     public static void main(String[] args) {
115
116         Matrix m = new Matrix();
117         System.out.println(m.header);
118         int M, N;
119     }
```

Question2.java

```
120      System.out.print("Enter the row size of the matrix: ");
121      N = in.nextInt();
122
123      System.out.print("Enter the column size of the matrix: ");
124      M = in.nextInt();
125
126      m.matrixAddition(N, M);
127      m.matrixMultiplation(N, M);
128      System.out.print("\n");
129  }
130 }
```