

Lesson 01 Demo 05

Implementing Loops in Java

Objective: To implement loops in Java

Tools required: Eclipse IDE

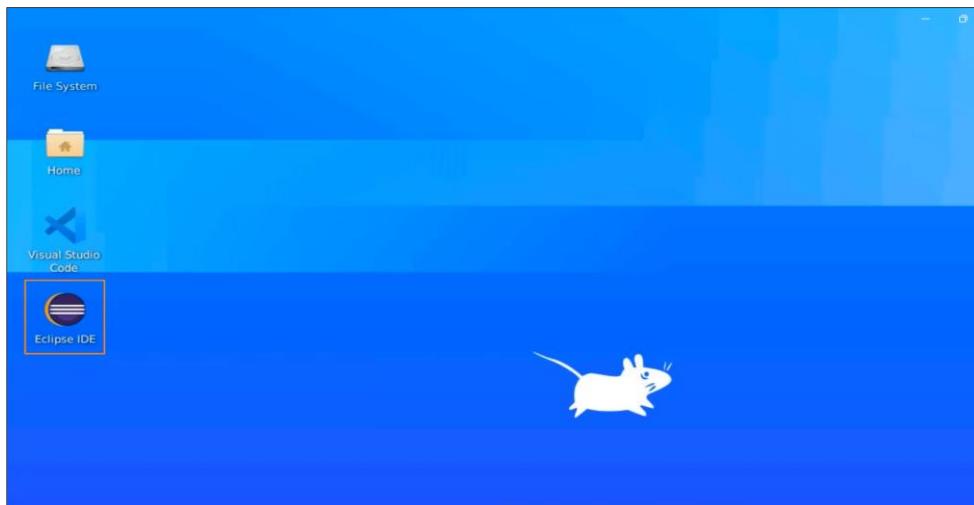
Prerequisites: None

Steps to be followed:

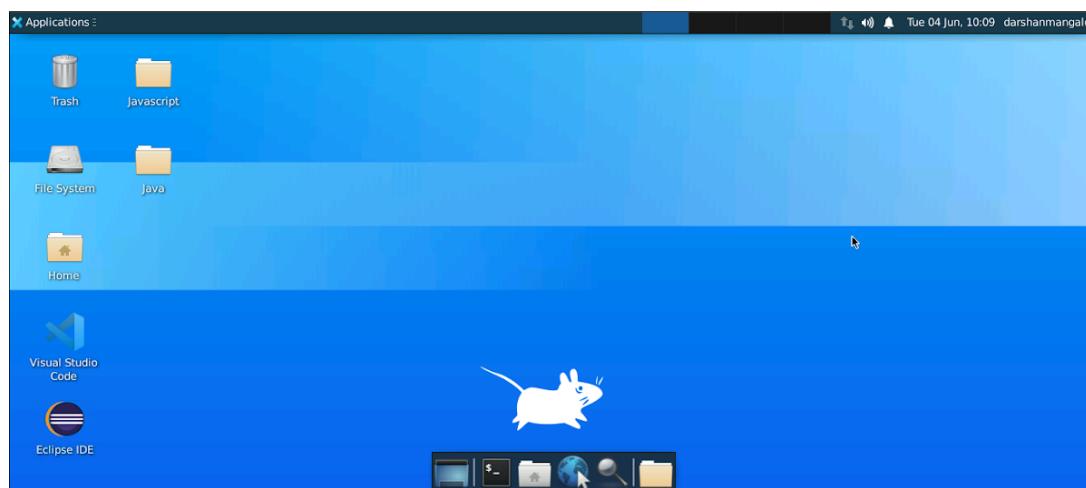
1. Create a folder on the Eclipse IDE
2. Create a program for the implementation of the Loops in Java

Step 1: Create a folder on the Eclipse IDE

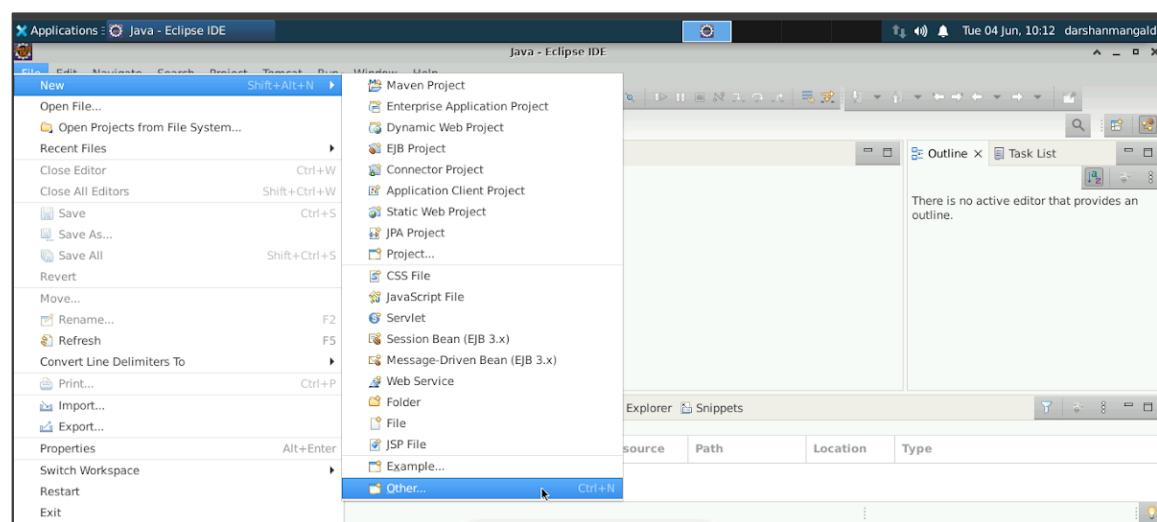
1.1 Open the Eclipse IDE



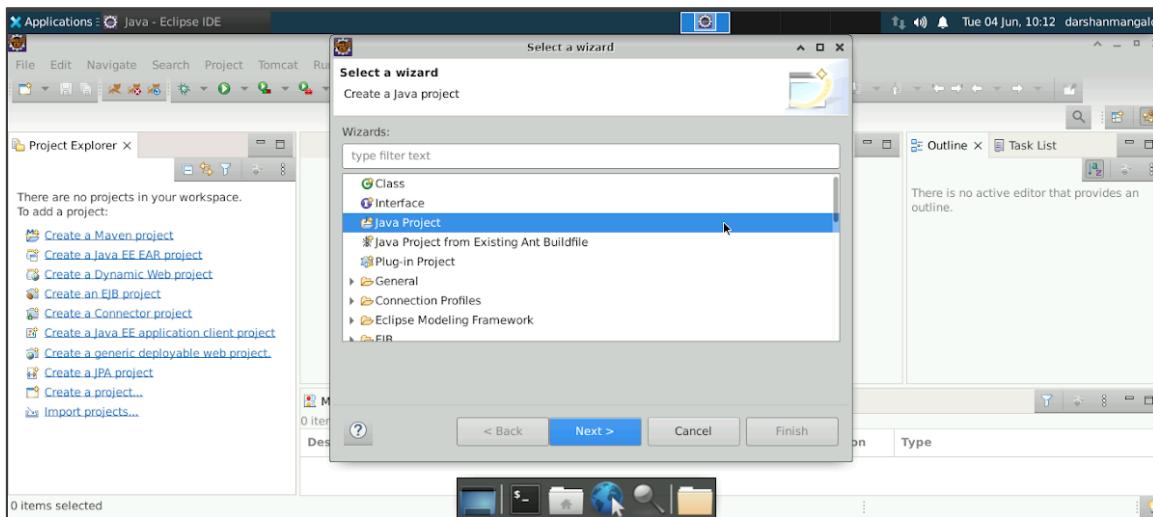
1.2. Create a folder on the desktop with any name of your choice let us say “**Java**” and open this folder with the **Eclipse IDE**



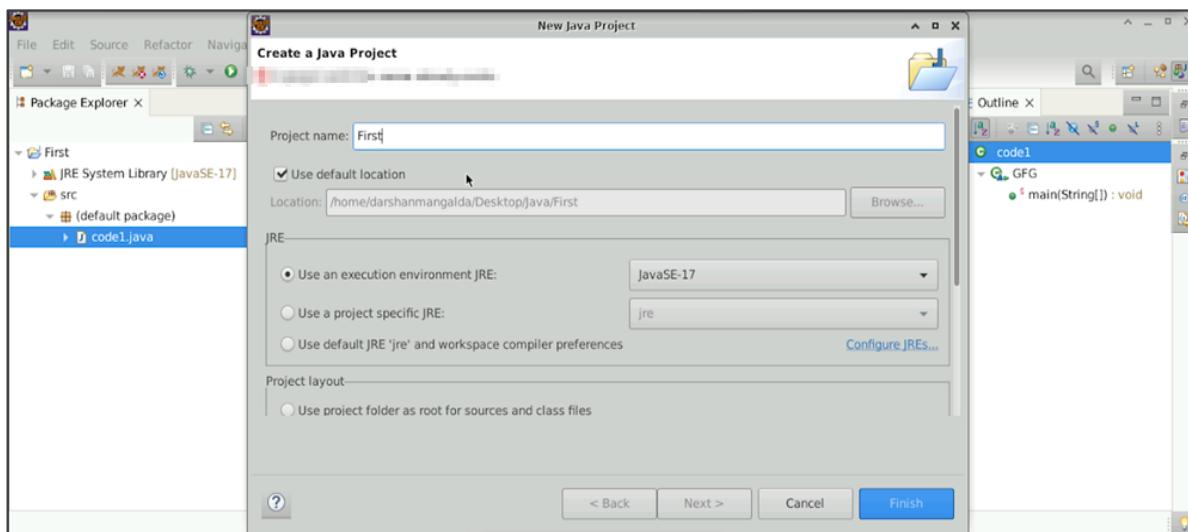
1.3. Select **File**, then click on **Other**



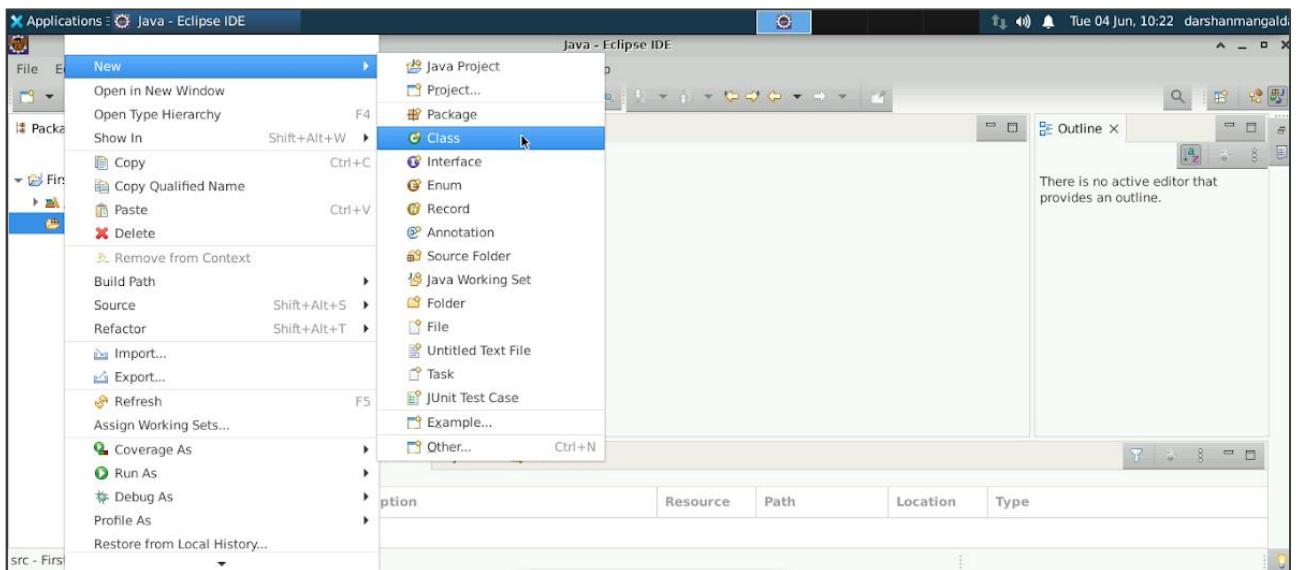
1.4 Select Java Project in the Select a Wizard tab



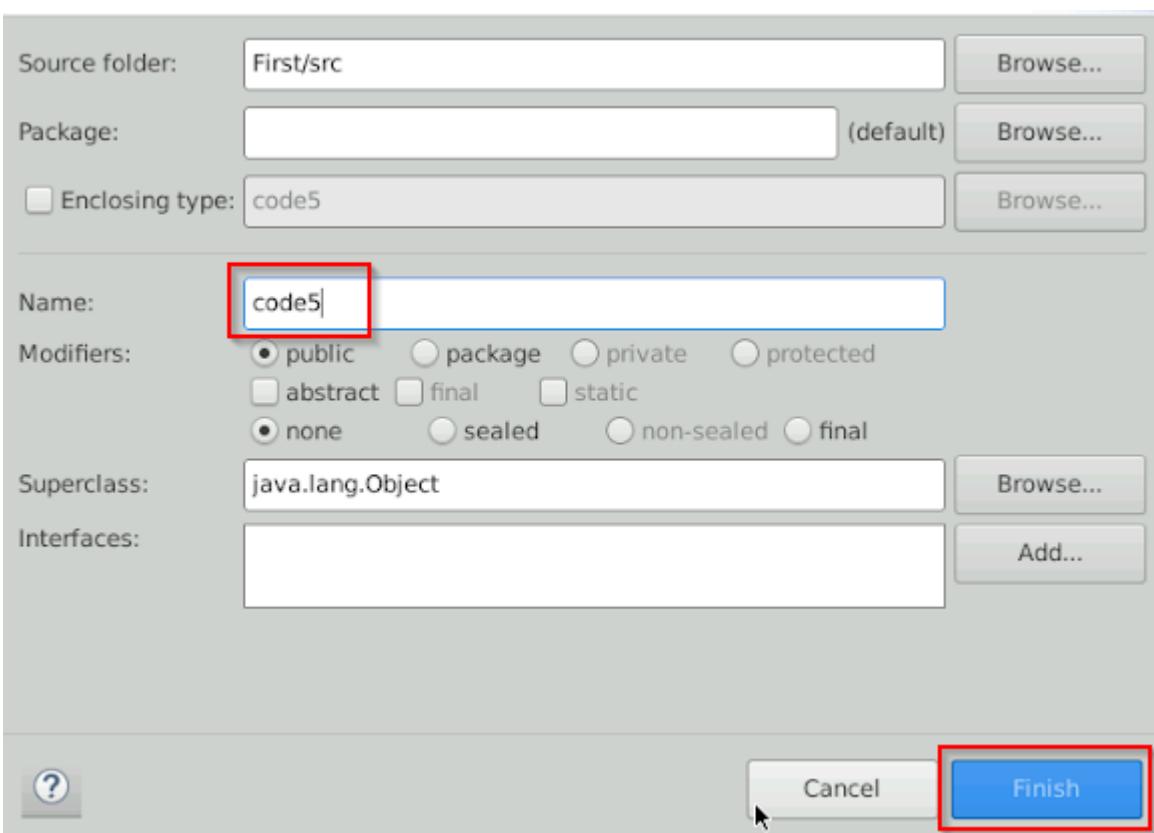
1.5 Name the Java Project as First in the Project name field and click on Finish Button



1.6 Right Src folder, select New, and click on the class option

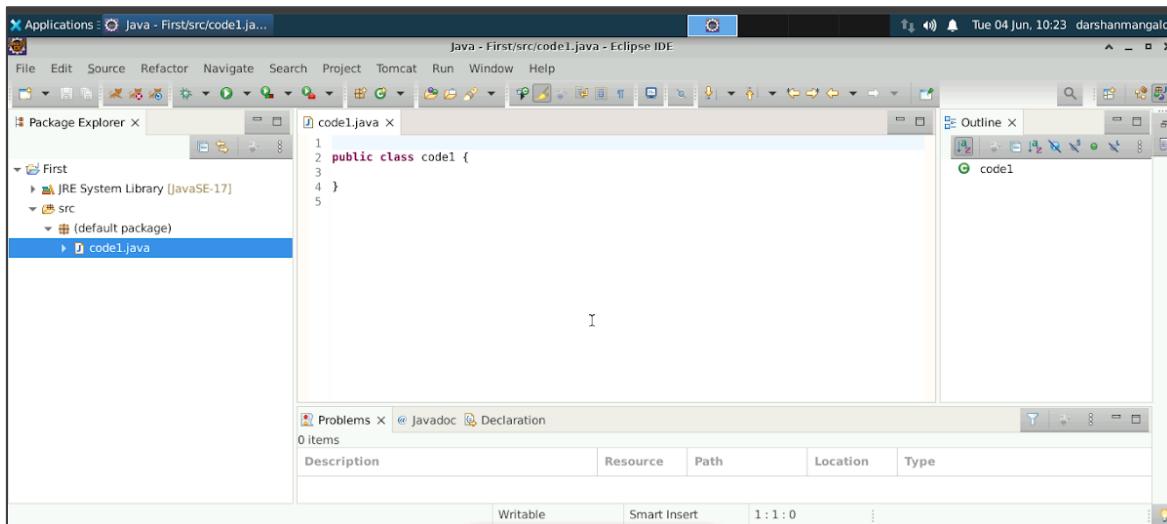


1.7 Enter the name in the Java class as **code5**, make sure the public option is checked, and click on **Finish** Button



Step 2: Create a program for the implementation of the Loops in Java

2.1 The Java file will be opened as shown below:

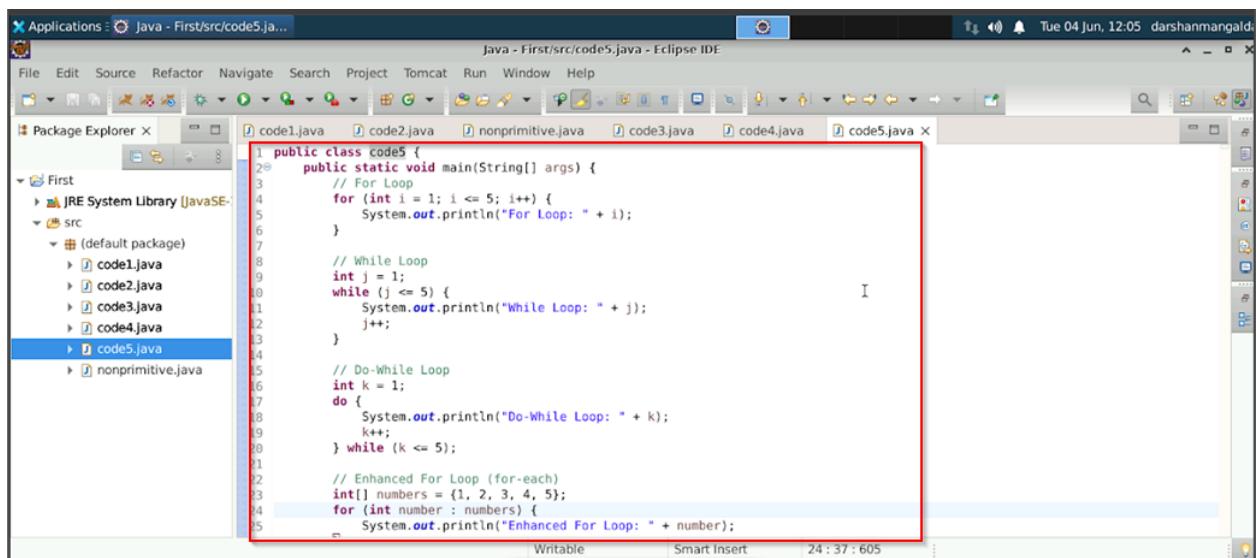


The screenshot shows the Eclipse IDE interface. The title bar reads "Java - First/src/code1.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Tomcat, Run, Window, Help. The Package Explorer view on the left shows a project named "First" with a JRE System Library [JavaSE-17] and a src folder containing code1.java. The central editor window displays the following Java code:

```
1 public class code1 {  
2     }  
3 }  
4 }
```

The Problems view at the bottom shows 0 items.

2.2 Enter the code for the loops in Java



The screenshot shows the Eclipse IDE interface with multiple Java files open in tabs: code1.java, code2.java, nonprimitive.java, code3.java, code4.java, and code5.java. The code5.java tab is active. The code in the editor is highlighted with a red box around the loop sections. The code is as follows:

```
1 public class code5 {  
2     public static void main(String[] args) {  
3         // For Loop  
4         for (int i = 1; i <= 5; i++) {  
5             System.out.println("For Loop: " + i);  
6         }  
7         // While Loop  
8         int j = 1;  
9         while (j <= 5) {  
10             System.out.println("While Loop: " + j);  
11             j++;  
12         }  
13         // Do-While Loop  
14         int k = 1;  
15         do {  
16             System.out.println("Do-While Loop: " + k);  
17             k++;  
18         } while (k <= 5);  
19         // Enhanced For Loop (for-each)  
20         int[] numbers = {1, 2, 3, 4, 5};  
21         for (int number : numbers) {  
22             System.out.println("Enhanced For Loop: " + number);  
23         }  
24     }  
25 }
```

The Problems view at the bottom shows 0 items.

Java - First/src/code5.java - Eclipse IDE

```

21 // Enhanced For Loop (for-each)
22 int[] numbers = {1, 2, 3, 4, 5};
23 for (int number : numbers) {
24     System.out.println("Enhanced For Loop: " + number);
25 }
26
27 // Nested For Loop
28 for (int m = 1; m <= 3; m++) {
29     for (int n = 1; n <= 3; n++) {
30         System.out.println("Nested For Loop: m = " + m + ", n = " + n);
31     }
32 }
33
34 // Infinite Loop (commented out to avoid infinite execution)
35 // for (;;) {
36 //     System.out.println("This is an infinite loop. Press Ctrl+C to stop.");
37 // }
38
39 // Break Example
40 for (int p = 1; p <= 10; p++) {
41     if (p == 5) {
42         break; // Exit the loop when p is 5
43     }
44 }
45 System.out.println("Break Example: " + p);

```

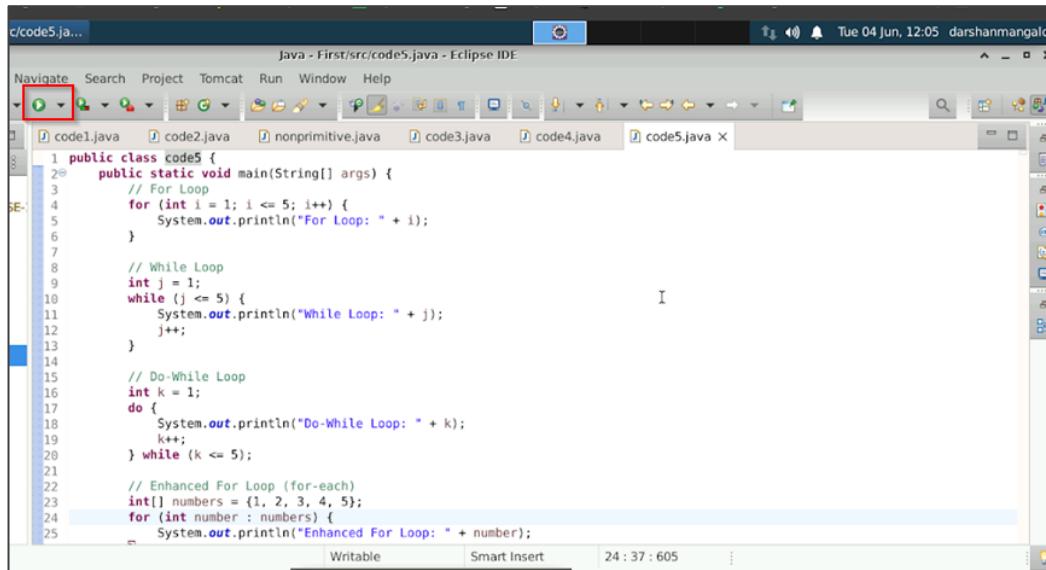
Java - First/src/code5.java - Eclipse IDE

```

33 }
34
35 // Infinite Loop (commented out to avoid infinite execution)
36 // for (;;) {
37 //     System.out.println("This is an infinite loop. Press Ctrl+C to stop.");
38 // }
39
40 // Break Example
41 for (int p = 1; p <= 10; p++) {
42     if (p == 5) {
43         break; // Exit the loop when p is 5
44     }
45     System.out.println("Break Example: " + p);
46 }
47
48 // Continue Example
49 for (int q = 1; q <= 10; q++) {
50     if (q == 5) {
51         continue; // Skip the iteration when q is 5
52     }
53     System.out.println("Continue Example: " + q);
54 }
55
56 }
57

```

2.3 Click on the Run Button to execute the code



```

public class code5 {
    public static void main(String[] args) {
        // For Loop
        for (int i = 1; i <= 5; i++) {
            System.out.println("For Loop: " + i);
        }

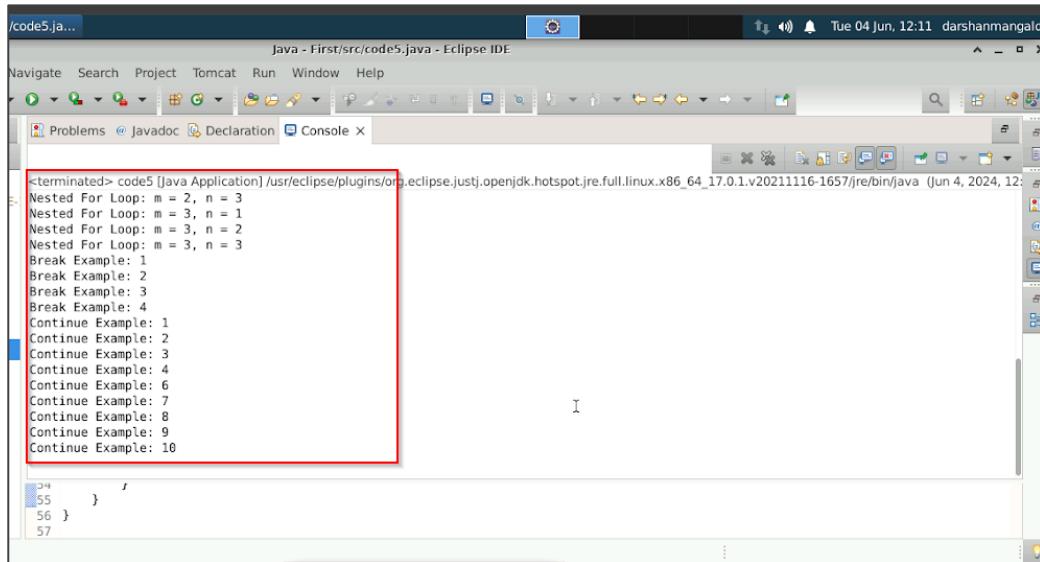
        // While Loop
        int j = 1;
        while (j <= 5) {
            System.out.println("While Loop: " + j);
            j++;
        }

        // Do-While Loop
        int k = 1;
        do {
            System.out.println("Do-While Loop: " + k);
            k++;
        } while (k <= 5);

        // Enhanced For Loop (for-each)
        int[] numbers = {1, 2, 3, 4, 5};
        for (int number : numbers) {
            System.out.println("Enhanced For Loop: " + number);
        }
    }
}

```

The output of the first Java program can be seen in the Console field as shown below:



```

<terminated> code5 [java Application] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.1.v20211116-1657/jre/bin/java (Jun 4, 2024, 12:11:11)
Nested For Loop: m = 2, n = 3
Nested For Loop: m = 3, n = 1
Nested For Loop: m = 3, n = 2
Nested For Loop: m = 3, n = 3
Break Example: 1
Break Example: 2
Break Example: 3
Break Example: 4
Continue Example: 1
Continue Example: 2
Continue Example: 3
Continue Example: 4
Continue Example: 6
Continue Example: 7
Continue Example: 8
Continue Example: 9
Continue Example: 10

```

By following the above steps, you have successfully implemented the loops concept in Java.