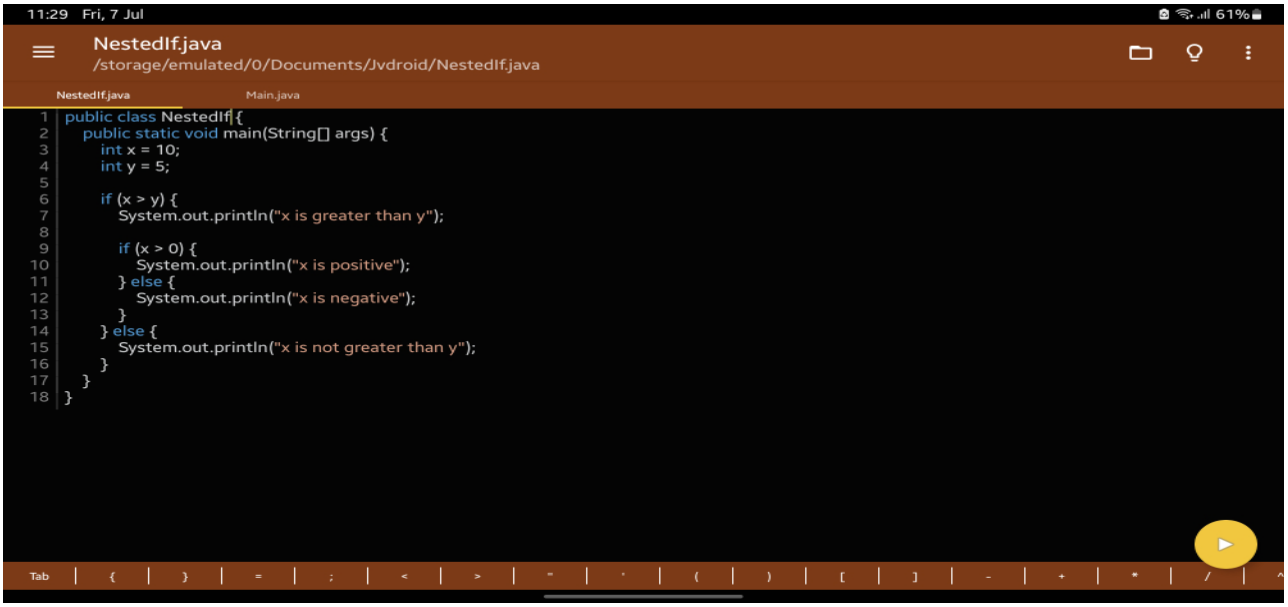
SHALINI LADWA

KODNEST CSR

1) Is nested simple if is permitted in Java?

Yes, nested if statements are permitted in Java. A nested if statement is an if statement that is located within the body of another if statement. It allows for more complex conditional branching based on multiple conditions.

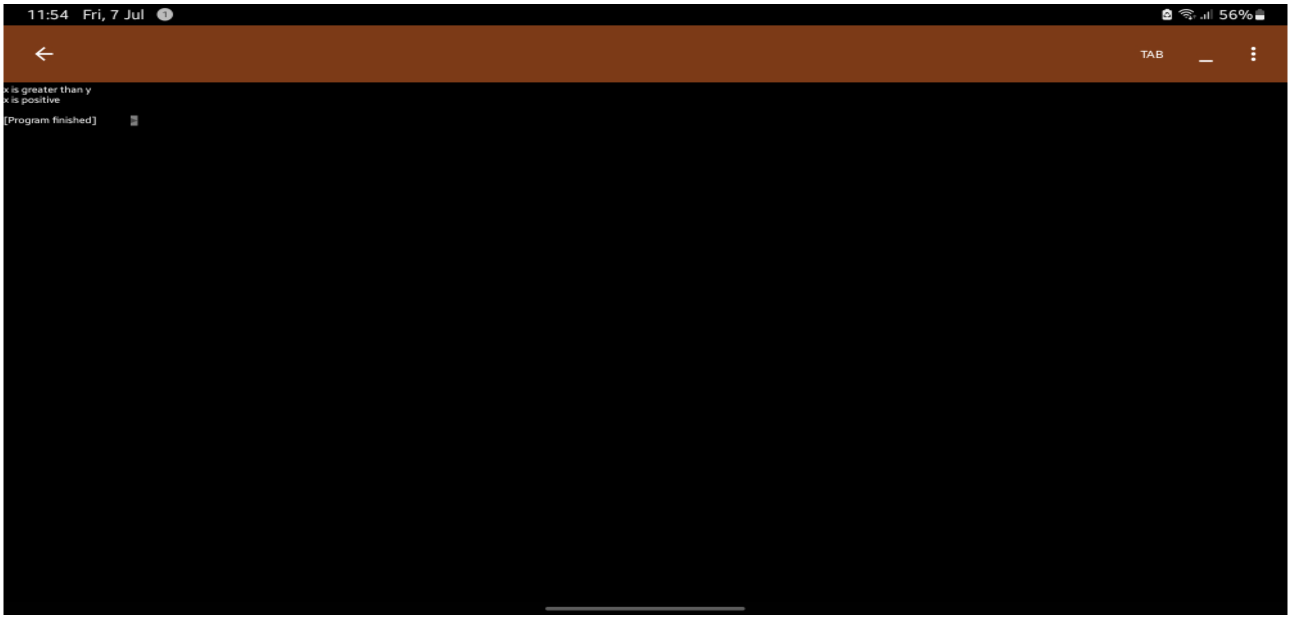
Example :


11:29 Fri, 7 Jul
61%
NestedIf.java
/storage/emulated/0/Documents/Jvdroid/NestedIf.java
NestedIf.java
Main.java
Tab { } = ; < > " ' ( ) [ ] - + * / ^

```java
public class NestedIf {
  public static void main(String[] args) {
    int x = 10;
    int y = 5;

    if (x > y) {
      System.out.println("x is greater than y");

      if (x > 0) {
        System.out.println("x is positive");
      } else {
        System.out.println("x is negative");
      }
    } else {
      System.out.println("x is not greater than y");
    }
  }
}
```



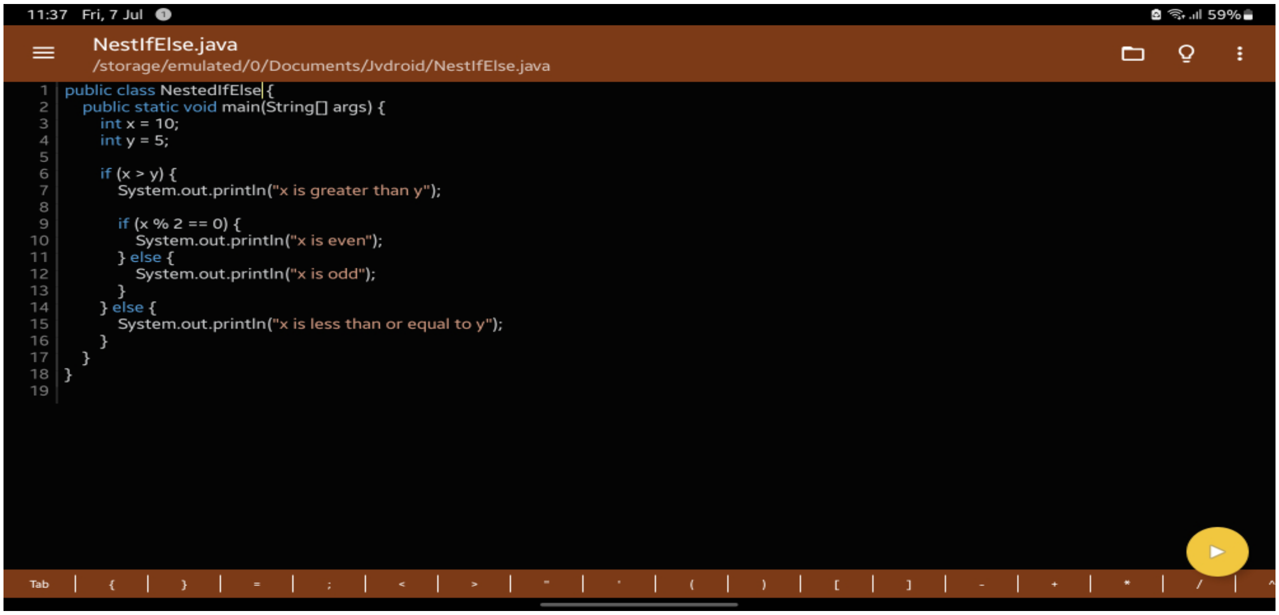
```
x is greater than y
x is positive

[Program finished]
```

In this example, there are two if statements. The first if statement checks if x is greater than y. If it is, the nested if statement inside the body of the first if statement checks if x is positive or non-positive. Based on the values of x and y, the appropriate messages are printed.

2) Is nested if else is permitted in Java?

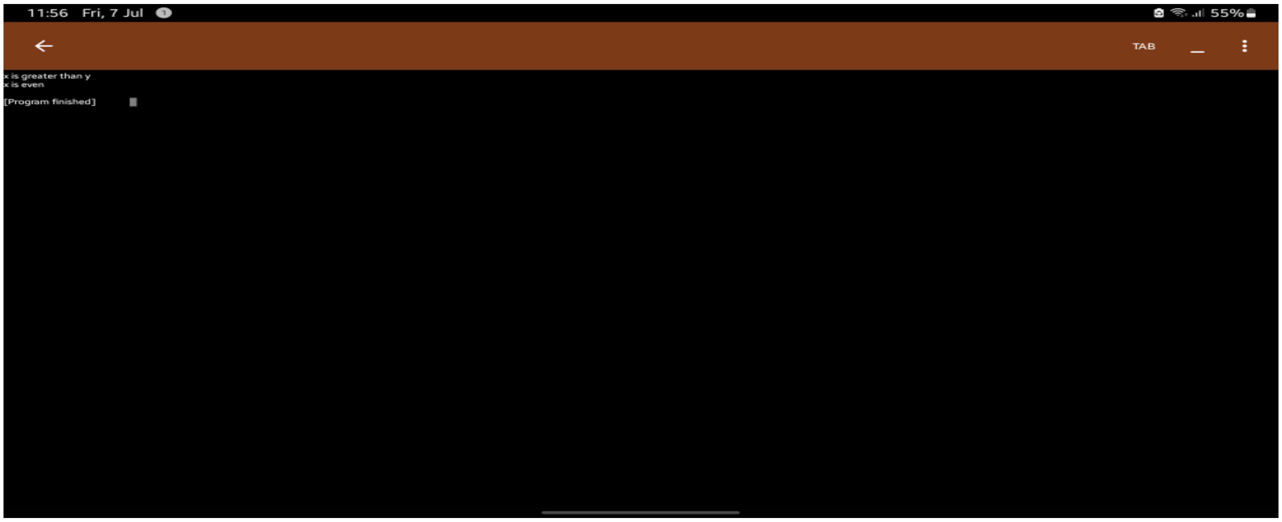
Yes, nested if-else statements are permitted in Java. In Java, you can nest if-else statements by placing an if-else statement inside another if or else block. This allows for more complex decision-making and branching within your code.


11:37 Fri, 7 Jul
59%
NestIfElse.java
/storage/emulated/0/Documents/Jvdroid/NestIfElse.java
Tab { } = ; < > " ' ( ) [ ] - + * /

```java
public class NestedIfElse {
  public static void main(String[] args) {
    int x = 10;
    int y = 5;

    if (x > y) {
      System.out.println("x is greater than y");

      if (x % 2 == 0) {
        System.out.println("x is even");
      } else {
        System.out.println("x is odd");
      }
    } else {
      System.out.println("x is less than or equal to y");
    }
  }
}
```

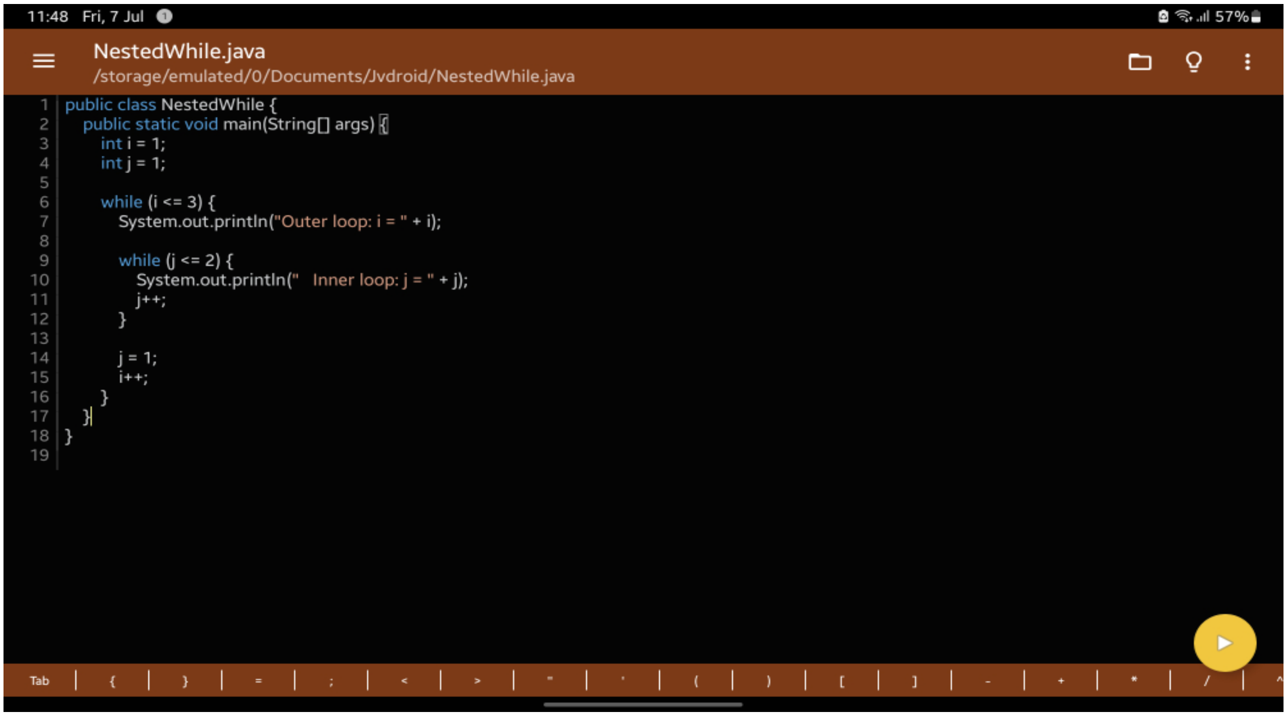


```
x is greater than y
x is even

[Program finished]
```

3) Is nested while is permitted in Java?

Yes, nested while loops are permitted in Java. You can have a while loop inside another while loop to create a nested loop structure. This allows for repeated execution of a block of code within another block of code.

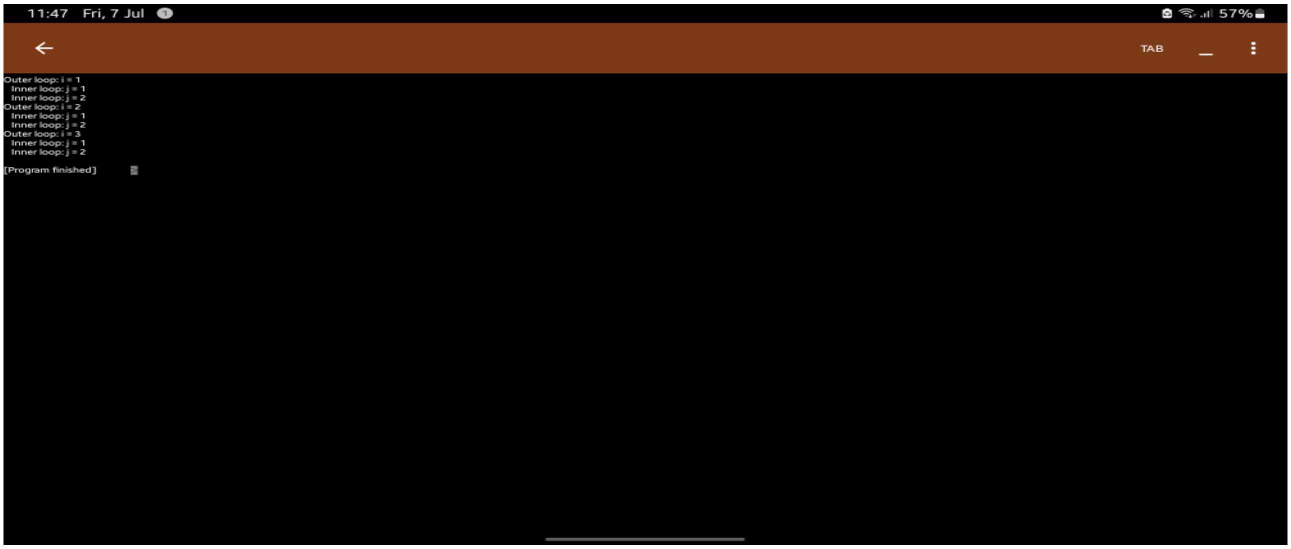

11:48 Fri, 7 Jul
57%
NestedWhile.java
/storage/emulated/0/Documents/Jvdroid/NestedWhile.java
Tab { } = ; < > " ' ( ) [ ] - + * / ^

```java
public class NestedWhile {
   public static void main(String[] args) {
      int i = 1;
      int j = 1;

      while (i <= 3) {
         System.out.println("Outer loop: i = " + i);

         while (j <= 2) {
            System.out.println("   Inner loop: j = " + j);
            j++;
         }

         j = 1;
         i++;
      }
   }
}
```

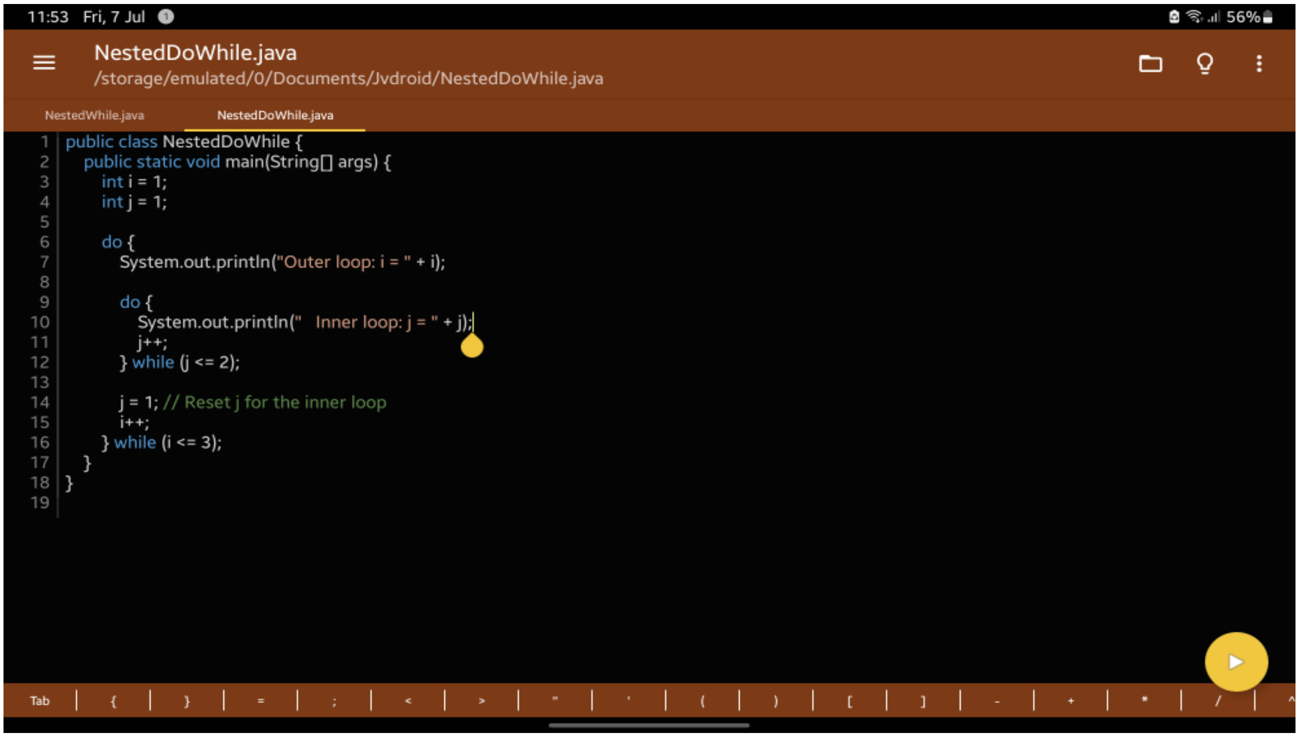


```
Outer loop: i = 1
  Inner loop: j = 1
  Inner loop: j = 2
Outer loop: i = 2
  Inner loop: j = 1
  Inner loop: j = 2
Outer loop: i = 3
  Inner loop: j = 1
  Inner loop: j = 2

[Program finished]
```

4)Is nested do-while is permitted in Java?

Yes, nested do-while loops are permitted in Java. You can have a do-while loop inside another do-while loop to create a nested loop structure. This allows for repeated execution of a block of code within another block of code, similar to nested while loops.

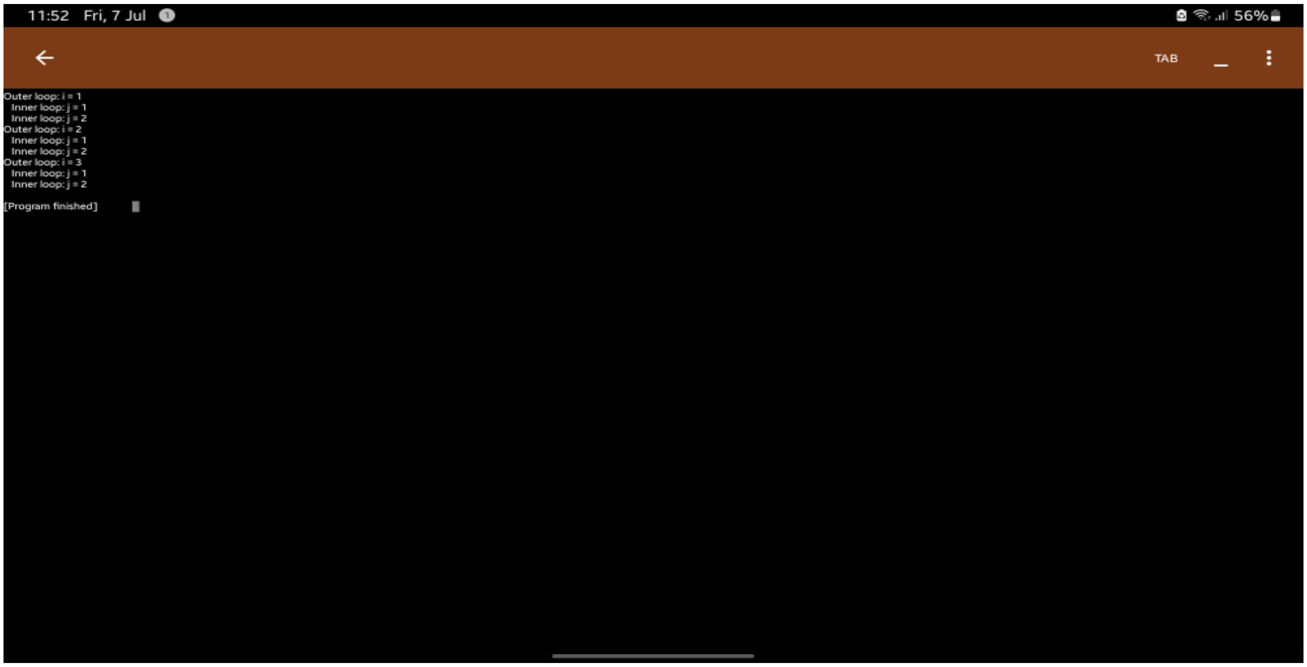

11:53 Fri, 7 Jul
56%
NestedDoWhile.java
/storage/emulated/0/Documents/Jvdroid/NestedDoWhile.java
NestedWhile.java
NestedDoWhile.java
Tab { } = ; < > " ' ( ) [ ] - + * /

```java
public class NestedDoWhile {
  public static void main(String[] args) {
    int i = 1;
    int j = 1;

    do {
      System.out.println("Outer loop: i = " + i);

      do {
        System.out.println("   Inner loop: j = " + j);
        j++;
      } while (j <= 2);

      j = 1; // Reset j for the inner loop
      i++;
    } while (i <= 3);
  }
}
```



```
Outer loop: i = 1
   Inner loop: j = 1
   Inner loop: j = 2
Outer loop: i = 2
   Inner loop: j = 1
   Inner loop: j = 2
Outer loop: i = 3
   Inner loop: j = 1
   Inner loop: j = 2

[Program finished]
```