WEEK END TASK
JAVA

CASE: 1

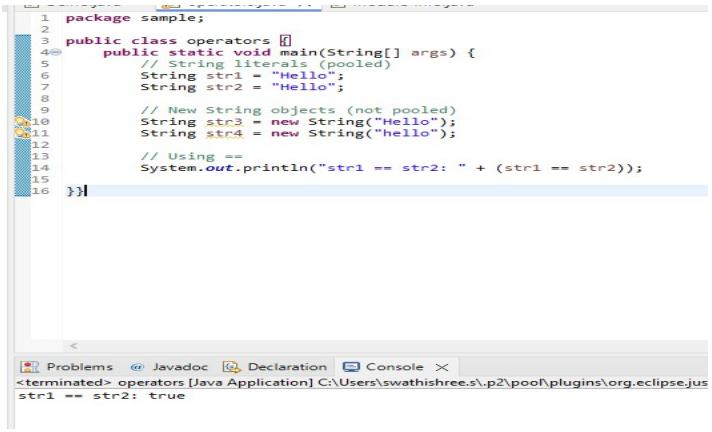
QUESTION: 1

```
public class StringComparisonExample {
   public static void main(String[] args) {
      // String literals (pooled)
      String str1 = "Hello";
      String str2 = "Hello";

      // New String objects (not pooled)
      String str3 = new String("Hello");
      String str4 = new String("hello");

      // Using ==
      System.out.println("str1 == str2: " + (str1 == str2)); // 1. (same memory reference) what's the result?
      System.out.println("str1 == str3: " + (str1 == str3)); //2. (different memory references) what's the result?

      // Using equals()
      System.out.println("str1.equals(str3): " + str1.equals(str3)); //3.  (same content) what's the result?
      System.out.println("str1.equals(str4): " + str1.equals(str4)); //4.  (case-sensitive) what's the result?

      // Using equalsIgnoreCase()
      System.out.println("str1.equalsIgnoreCase(str4): " + str1.equalsIgnoreCase(str4)); //5. (case-insensitive) what's the result?
   }
}
```

ANSWER:

1) str1 == str2:

Result: True



2) str1 == str3:

Result: False

```java
package sample;

public class operators {
    public static void main(String[] args) {
        // String literals (pooled)
        String str1 = "Hello";
        String str2 = "Hello";

        // New String objects (not pooled)
        String str3 = new String("Hello");
        String str4 = new String("hello");

        // Using ==
        System.out.println("str1 == str3: " + (str1 == str3));

}}
```

Problems @ Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.open
str1 == str3: false
```

3) str1.equals(str3):

Result: True

```java
package sample;

public class operators {
    public static void main(String[] args) {
        // String literals (pooled)
        String str1 = "Hello";
        String str2 = "Hello";

        // New String objects (not pooled)
        String str3 = new String("Hello");
        String str4 = new String("hello");

     // Using equals()
        System.out.println("str1.equals(str3): " + str1.equals(str3));

}}
```

Problems @ Javadoc Declaration Console

```
erminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre
tr1.equals(str3): true
```

4) str1.equals(str4):

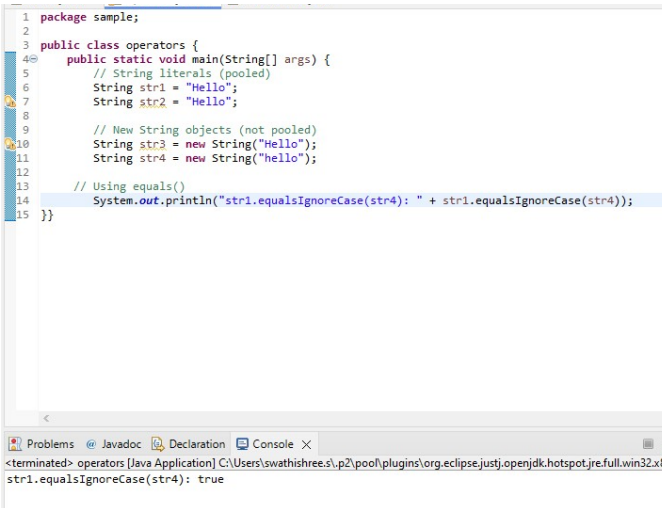
Result: False

```java
package sample;

public class operators {
    public static void main(String[] args) {
        // String literals (pooled)
        String str1 = "Hello";
        String str2 = "Hello";

        // New String objects (not pooled)
        String str3 = new String("Hello");
        String str4 = new String("hello");

     // Using equals()
        System.out.println("str1.equals(str4): " + str1.equals(str4));
}}
```

Problems @ Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.h
str1.equals(str4): false
```

5) str1.equalsIgnoreCase(str4):
Result: True



CASE:2

QUESTION:2

```
public class IntegerComparisonExample {
   public static void main(String[] args) {

//Mention what's the result in 1, 2, 3,4 and 5
      // Primitive int
      int int1 = 100;
      int int2 = 100;

      // Integer objects
      Integer intObj1 = 100;
      Integer intObj2 = 100;
      Integer intObj3 = new Integer(100);
      Integer intObj4 = new Integer(200);

      // Using == with primitive int
      System.out.println("int1 == int2: " + (int1 == int2)); // 1. (compares values)

      // Using == with Integer objects (within -128 to 127 range)
      System.out.println("intObj1 == intObj2: " + (intObj1 == intObj2)); // 2. (cached objects)

      // Using == with Integer objects (new instance)
      System.out.println("intObj1 == intObj3: " + (intObj1 == intObj3)); // 3. (different
instances)

      // Using equals() with Integer objects
      System.out.println("intObj1.equals(intObj3): " + intObj1.equals(intObj3)); // 4. (same
content)
      System.out.println("intObj1.equals(intObj4): " + intObj1.equals(intObj4)); // 5.
(different content)
```

```
    }
}
```

ANSWER:

1) int1 == int2:

Result: True

```java
package sample;

public class operators {
    public static void main(String[] args) {

//Mention what's the result in 1, 2, 3,4 and 5
        // Primitive int
        int int1 = 100;
        int int2 = 100;

        // Integer objects
        Integer intObj1 = 100;
        Integer intObj2 = 100;
        Integer intObj3 = new Integer(100);
        Integer intObj4 = new Integer(200);

        // Using == with primitive int
        System.out.println("int1 == int2: " + (int1 == int2)); // 1. (compares values)
}}
```

Problems Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.
int1 == int2: true
```

2) intObj1 == intObj2:

Result: True

```java
package sample;

public class operators {
    public static void main(String[] args) {

//Mention what's the result in 1, 2, 3,4 and 5
        // Primitive int
        int int1 = 100;
        int int2 = 100;

        // Integer objects
        Integer intObj1 = 100;
        Integer intObj2 = 100;
        Integer intObj3 = new Integer(100);
        Integer intObj4 = new Integer(200);

        // Using == with Integer objects (within -128 to 127 range)
        System.out.println("intObj1 == intObj2: " + (intObj1 == intObj2));
}}
```

Problems Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspo
intObj1 == intObj2: true
```

3) intObj1 == intObj3:

Result: False

```java
package sample;

public class operators {
    public static void main(String[] args) {

//Mention what's the result in 1, 2, 3,4 and 5
        // Primitive int
        int int1 = 100;
        int int2 = 100;

        // Integer objects
        Integer intObj1 = 100;
        Integer intObj2 = 100;
        Integer intObj3 = new Integer(100);
        Integer intObj4 = new Integer(200);


        // Using == with Integer objects (new instance)
        System.out.println("intObj1 == intObj3: " + (intObj1 == intObj3));
}}
```

Problems @ Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.hot
intObj1 == intObj3: false
```

4) intObj1.equals(intObj3):

Result: True

```java
package sample;

public class operators {
    public static void main(String[] args) {

//Mention what's the result in 1, 2, 3,4 and 5
        // Primitive int
        int int1 = 100;
        int int2 = 100;

        // Integer objects
        Integer intObj1 = 100;
        Integer intObj2 = 100;
        Integer intObj3 = new Integer(100);
        Integer intObj4 = new Integer(200);


        // Using equals() with Integer objects
        System.out.println("intObj1.equals(intObj3): " + intObj1.equals(intObj3));
}}
```

Problems @ Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.
intObj1.equals(intObj3): true
```

5) intObj1.equals(intObj4):

Result: False

```
package sample;

public class operators {
    public static void main(String[] args) {

//Mention what's the result in 1, 2, 3,4 and 5
        // Primitive int
        int int1 = 100;
        int int2 = 100;

        // Integer objects
        Integer intObj1 = 100;
        Integer intObj2 = 100;
        Integer intObj3 = new Integer(100);
        Integer intObj4 = new Integer(200);


        System.out.println("intObj1.equals(intObj4): " + intObj1.equals(intObj4));
}}
```

Problems Javadoc Declaration Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.
intObj1.equals(intObj4): false
```

CASE: 3

QUESTION: 3

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class TryWithResourcesExample {
//Eliminating finally block to close resources.

    public static void main(String[] args) {
        // File path (adjust the path as needed)
        String filePath = "example.txt";

        // Traditional try-with-resources block
        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
```

```
        e.printStackTrace();
    }
  }
}
```

ANSWER:

Automatic Resource Management:

The BufferedReader and FileReader resources are automatically closed when the try-with-resources block is exited. You do not need a finally block to close these resources manually, which is the traditional approach.

Enhanced Code (Using try-with-resources):

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class TryWithResourcesExample {
//Eliminating finally block to close resources.
public static void main(String[] args) {
// File path (adjust the path as needed)
String filePath = "example.txt";
try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
String line;
while ((line = reader.readLine()) != null) {
System.out.println(line);
}
} catch (IOException e) {
e.printStackTrace();
}
}
}
```

```
13      try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
14        String line;
15        while ((line = reader.readLine()) != null) {
16            System.out.println(line);
17        }
18    } catch (IOException e) {
19        e.printStackTrace();
20    }
21     }
22    }
23
24
```

Problems Javadoc Declaration Console

```
<terminated> CopyBytes [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (11 Aug 2024, 11:18:33 am – 11:
Hello everyone!,Welcome to Java
```

CASE: 4

QUESTION: 4

```
import java.util.HashSet;
import java.util.LinkedHashSet;
import java.util.Set;
import java.util.TreeSet;

public class SetExample {
   public static void main(String[] args) {
      // Set 1. What's the order of elements?
      Set<String> hashSet = new HashSet<>();
      hashSet.add("Banana");
      hashSet.add("Apple");
      hashSet.add("Orange");
      hashSet.add("Grapes");

      System.out.println("HashSet: " + hashSet);

      // LinkedHashSet 2. What's the order of elements  ?
      Set<String> linkedHashSet = new LinkedHashSet<>();
      linkedHashSet.add("Banana");
      linkedHashSet.add("Apple");
      linkedHashSet.add("Orange");
      linkedHashSet.add("Grapes");
```

```
        System.out.println("LinkedHashSet: " + linkedHashSet);

        // TreeSet 1. What's the order of elements  ?
        Set<String> treeSet = new TreeSet<>();
        treeSet.add("Banana");
        treeSet.add("Apple");
        treeSet.add("Orange");
        treeSet.add("Grapes");

        System.out.println("TreeSet: " + treeSet);
    }
}
```

ANSWER:

Demo.java | operators.java | module-info.java

```
        Set<String> hashSet = new HashSet<>();
        hashSet.add("Banana");
        hashSet.add("Apple");
        hashSet.add("Orange");
        hashSet.add("Grapes");

        System.out.println("HashSet: " + hashSet);

        // LinkedHashSet 2. What's the order of elements  ?
        Set<String> linkedHashSet = new LinkedHashSet<>();
        linkedHashSet.add("Banana");
        linkedHashSet.add("Apple");
        linkedHashSet.add("Orange");
        linkedHashSet.add("Grapes");

        System.out.println("LinkedHashSet: " + linkedHashSet);

        // TreeSet 1. What's the order of elements  ?
        Set<String> treeSet = new TreeSet<>();
        treeSet.add("Banana");
        treeSet.add("Apple");
        treeSet.add("Orange");
        treeSet.add("Grapes");

        System.out.println("TreeSet: " + treeSet);
    }
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> operators [Java Application] C:\Users\swathishree.s\.p2\pool\plugins\org.eclipse.justj.openj
HashSet: [Apple, Grapes, Orange, Banana]
LinkedHashSet: [Banana, Apple, Orange, Grapes]
TreeSet: [Apple, Banana, Grapes, Orange]
```