

## Weekend Assignment in Java

1. Please find case 1 and mention the result for the mentioned statements using strings.

### CODE:

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

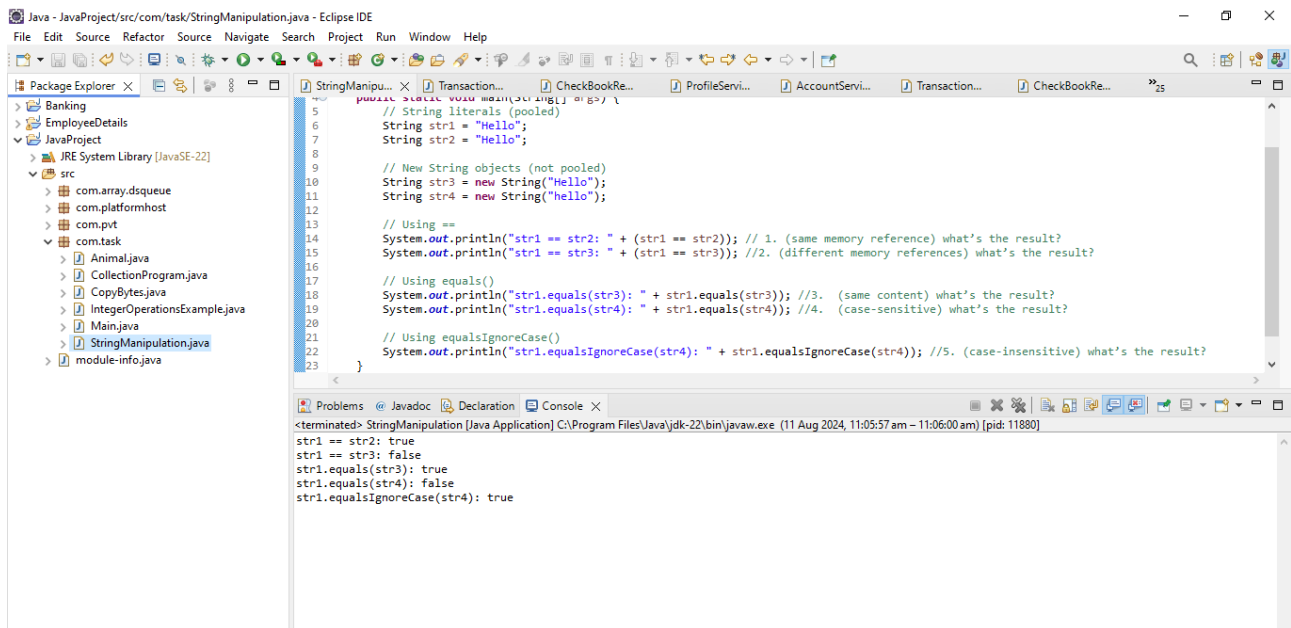
        // New String objects (not pooled)
        String str3 = new String("Hello");
        String str4 = new String("hello");

        // 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?
    }
}
```

## OUTPUT:



The screenshot shows the Eclipse IDE with a Java project named 'JavaProject'. The Package Explorer on the left shows the project structure, including 'src' and 'module-info.java'. The main editor displays 'StringManipulation.java' with the following code:

```
5 public static void main(String[] args) {\n6     // String literals (pooled)\n7     String str1 = "Hello";\n8     String str2 = "Hello";\n9     // New String objects (not pooled)\n10    String str3 = new String("Hello");\n11    String str4 = new String("hello");\n12\n13    // Using ==\n14    System.out.println("str1 == str2: " + (str1 == str2)); // 1. (same memory reference) what's the result?\n15    System.out.println("str1 == str3: " + (str1 == str3)); // 2. (different memory references) what's the result?\n16\n17    // Using equals()\n18    System.out.println("str1.equals(str3): " + str1.equals(str3)); // 3. (same content) what's the result?\n19    System.out.println("str1.equals(str4): " + str1.equals(str4)); // 4. (case-sensitive) what's the result?\n20\n21    // Using equalsIgnoreCase()\n22    System.out.println("str1.equalsIgnoreCase(str4): " + str1.equalsIgnoreCase(str4)); // 5. (case-insensitive) what's the result?\n23 }
```

The Console window at the bottom shows the output of the program:

```
<terminated> StringManipulation [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe. (11 Aug 2024, 11:05:57 am - 11:06:00 am) [pid: 11880]\nstr1 == str2: true\nstr1 == str3: false\nstr1.equals(str3): true\nstr1.equals(str4): false\nstr1.equalsIgnoreCase(str4): true
```

2. Find case 2 and mention the result for the statements using integers.

## CODE:

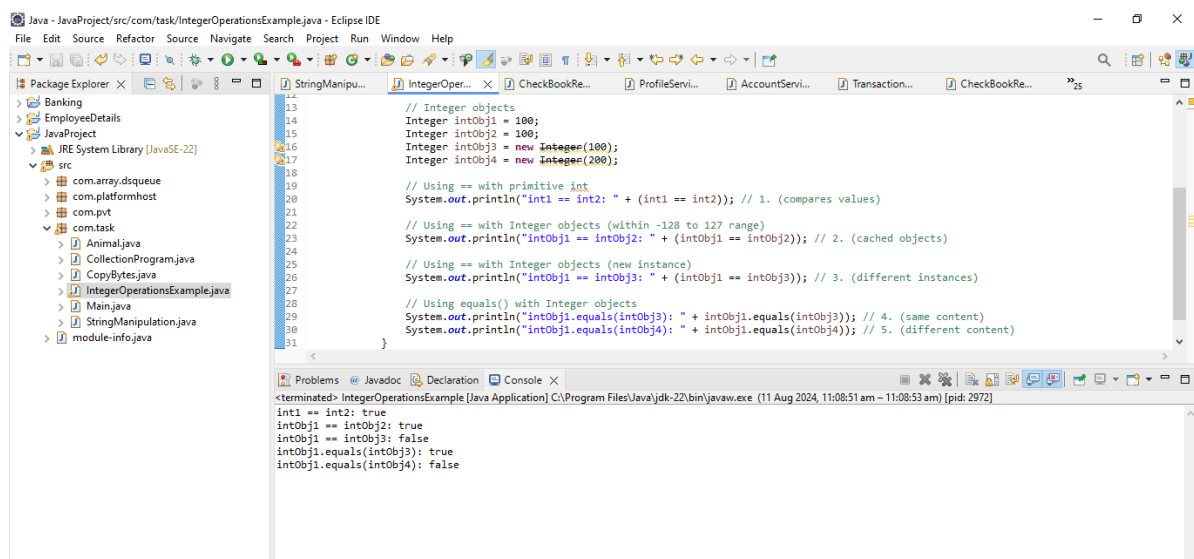
```
public class IntegerComparisonExample {\n    public static void main(String[] args) {\n        //Mention what's the result in 1, 2, 3,4 and 5\n        // Primitive int\n        int int1 = 100;\n        int int2 = 100;\n        // Integer objects\n        Integer intObj1 = 100;\n        Integer intObj2 = 100;\n        Integer intObj3 = new Integer(100);\n        Integer intObj4 = new Integer(200);\n\n        // Using == with primitive int\n        System.out.println("int1 == int2: " + (int1 == int2)); // 1. (compares values)\n\n        // Using == with Integer objects (within -128 to 127 range)\n        System.out.println("intObj1 == intObj2: " + (intObj1 == intObj2)); // 2. (cached\n        objects)\n\n        // 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)  
}  
}
```

## OUTPUT:



```
Java - JavaProject/src/com/task/IntegerOperationsExample.java - Eclipse IDE  
File Edit Source Refactor Source Navigate Search Project Run Window Help  
Package Explorer | StringManipu... | IntegerOper... | CheckBookRe... | ProfileServi... | AccountServi... | Transaction... | CheckBookRe... | 25  
> Banking  
> EmployeeDetails  
> JavaProject  
  > JRE System Library [JavaSE-22]  
  > src  
    > com.array.dsqueue  
    > com.platformhost  
    > com.pvt  
    > com.task  
      > Animal.java  
      > CollectionProgram.java  
      > CopyBytes.java  
      > IntegerOperationsExample.java  
      > Main.java  
      > StringManipulation.java  
      > module-info.java  
13 // Integer objects  
14 Integer intObj1 = 100;  
15 Integer intObj2 = 100;  
16 Integer intObj3 = new Integer(100);  
17 Integer intObj4 = new Integer(200);  
18  
19 // Using == with primitive int  
20 System.out.println("int1 == int2: " + (int1 == int2)); // 1. (compares values)  
21  
22 // Using == with Integer objects (within -128 to 127 range)  
23 System.out.println("intObj1 == intObj2: " + (intObj1 == intObj2)); // 2. (cached objects)  
24  
25 // Using == with Integer objects (new instance)  
26 System.out.println("intObj1 == intObj3: " + (intObj1 == intObj3)); // 3. (different instances)  
27  
28 // Using equals() with Integer objects  
29 System.out.println("intObj1.equals(intObj3): " + intObj1.equals(intObj3)); // 4. (same content)  
30 System.out.println("intObj1.equals(intObj4): " + intObj1.equals(intObj4)); // 5. (different content)  
31  
32 }  
Problems | Javadoc | Declaration | Console | X  
<terminated> IntegerOperationsExample [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (11 Aug 2024, 11:08:53 am) [pid: 2972]  
int1 == int2: true  
intObj1 == intObj2: true  
intObj1 == intObj3: false  
intObj1.equals(intObj3): true  
intObj1.equals(intObj4): false
```

3. Find case 3 and mention how Basic I/O resources are getting closed and the difference that you implemented earlier in the code – copyBytes.java

## CODE:

```
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):

```

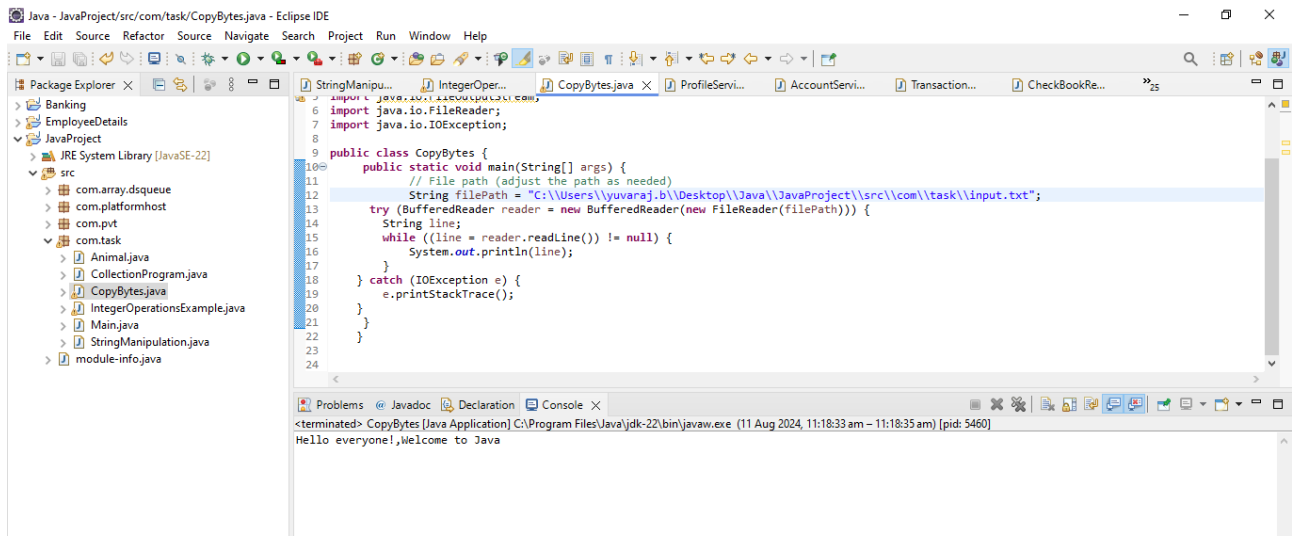
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

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();
        }
    }
}

```

## OUTPUT:



4. Find case 4 and mention the order for 1, 2 and 3 using collections

## CODE:

```
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");
```

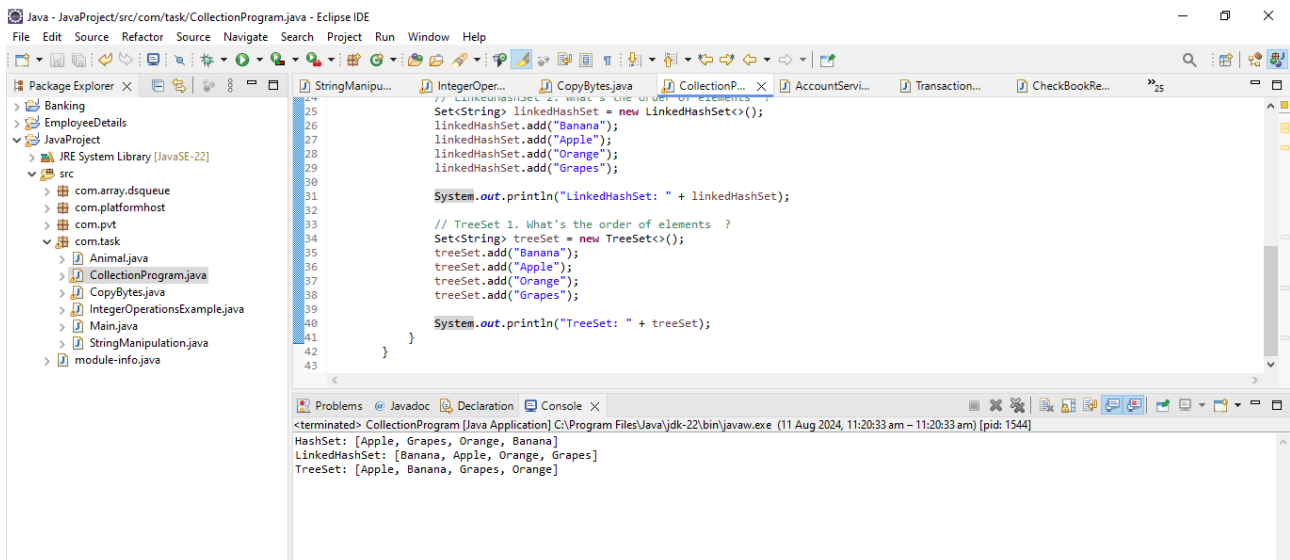
```
treeSet.add("Orange");
```

```
treeSet.add("Grapes");
```

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

```
    }  
}
```

## OUTPUT:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a package named 'com.task' containing several Java files, including 'CollectionProgram.java'. The main editor window shows the code for 'CollectionProgram.java', which includes the creation of a 'HashSet' and a 'TreeSet', both containing the elements 'Apple', 'Banana', 'Orange', and 'Grapes'. The console at the bottom shows the output of the program, displaying the elements in the order they were added to each set: 'HashSet: [Apple, Grapes, Orange, Banana]' and 'TreeSet: [Apple, Banana, Grapes, Orange]'.

```
Java - JavaProject/src/com/task/CollectionProgram.java - Eclipse IDE  
File Edit Source Refactor Source Navigate Search Project Run Window Help  
Package Explorer X  
Banking  
EmployeeDetails  
JavaProject  
JRE System Library [JavaSE-22]  
src  
com.array.dsqueue  
com.platformhost  
com.pvt  
com.task  
Animal.java  
CollectionProgram.java  
CopyBytes.java  
IntegerOperationsExample.java  
Main.java  
StringManipulation.java  
module-info.java  
StringManipu... IntegerOper... CopyBytes.java CollectionP... AccountServi... Transaction... CheckBookRe... 25  
25 Set<String> linkedHashSet = new LinkedHashSet<>();  
26 linkedHashSet.add("Banana");  
27 linkedHashSet.add("Apple");  
28 linkedHashSet.add("Orange");  
29 linkedHashSet.add("Grapes");  
30  
31 System.out.println("LinkedHashSet: " + linkedHashSet);  
32  
33 // TreeSet 1. What's the order of elements ?  
34 Set<String> treeSet = new TreeSet<>();  
35 treeSet.add("Banana");  
36 treeSet.add("Apple");  
37 treeSet.add("Orange");  
38 treeSet.add("Grapes");  
39  
40 System.out.println("TreeSet: " + treeSet);  
41 }  
42 }  
43  
Problems Javadoc Declaration Console X  
<terminated> CollectionProgram [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (11 Aug 2024, 11:20:33 am - 11:20:33 am) [pid: 1544]  
HashSet: [Apple, Grapes, Orange, Banana]  
LinkedHashSet: [Banana, Apple, Orange, Grapes]  
TreeSet: [Apple, Banana, Grapes, Orange]
```