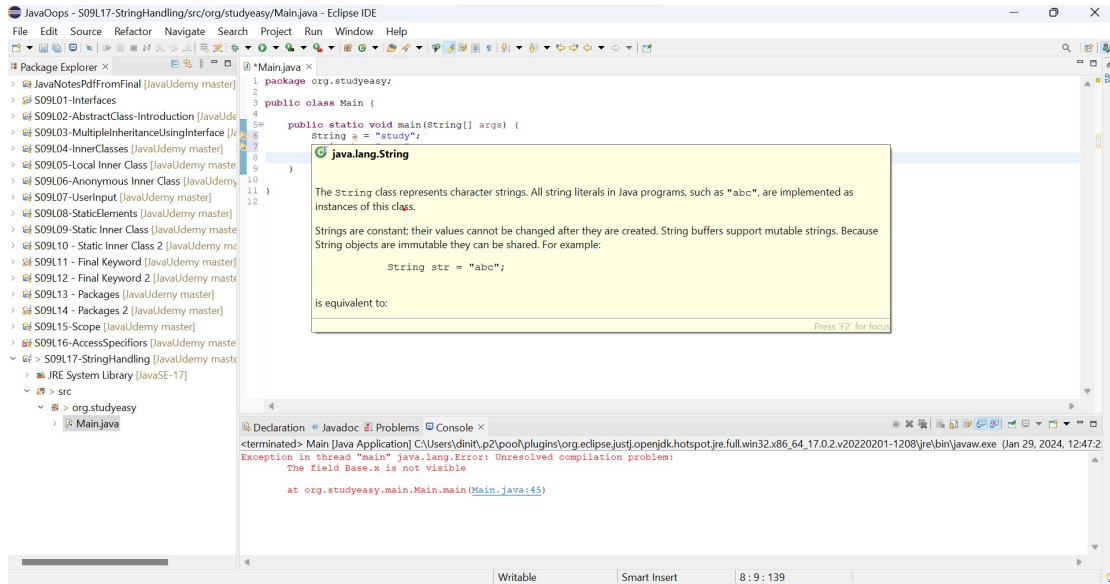
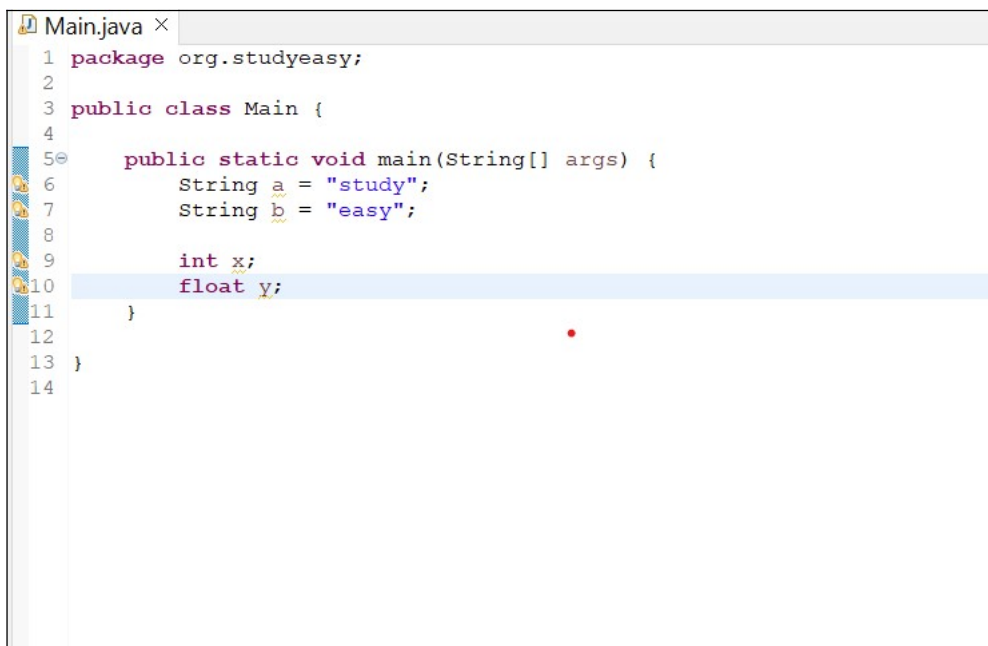


● STRING HANDLING

- String is a special variable or special datatype.
- Example : Hover over the String, you get popover.



- Why Java String is a Class (Difference between primitive and non primitive datatype)?



The above pic is the difference between primitive datatype and non primitive datatype.

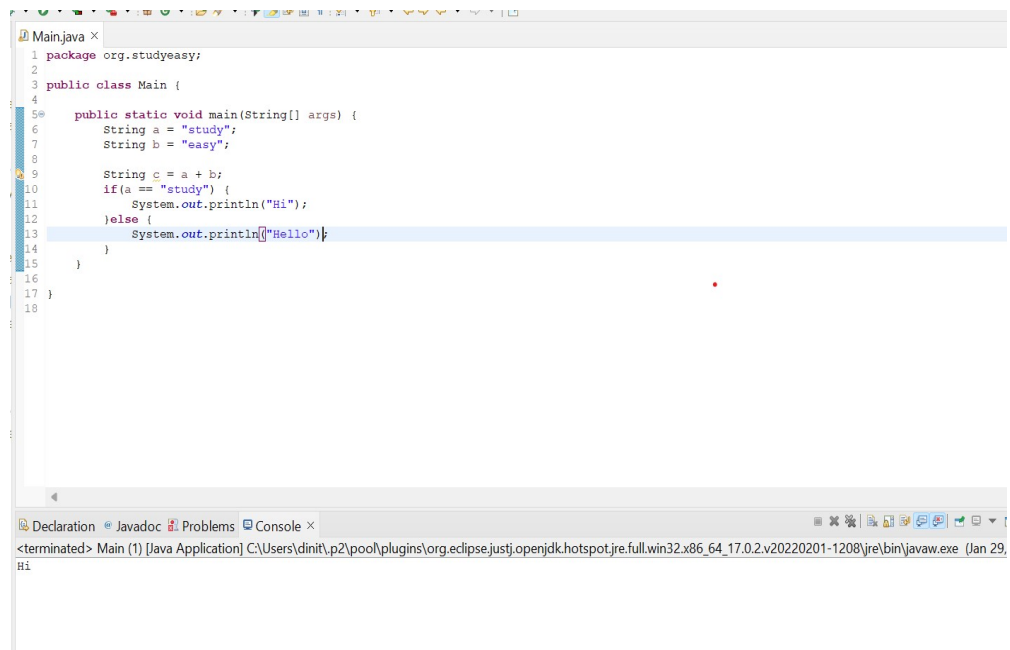
- Why Java String is a Class? '+' as Overloaded operator - Can add both primitive and objects.



The screenshot shows the Eclipse IDE with a file named 'Main.java'. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String a = "study";
7         String b = "easy";
8
9         String c = a + b;
10        System.out.println(c);
11    }
12 }
13
14
```

The console output at the bottom reads: <terminated> Main (1) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32-studyeasy

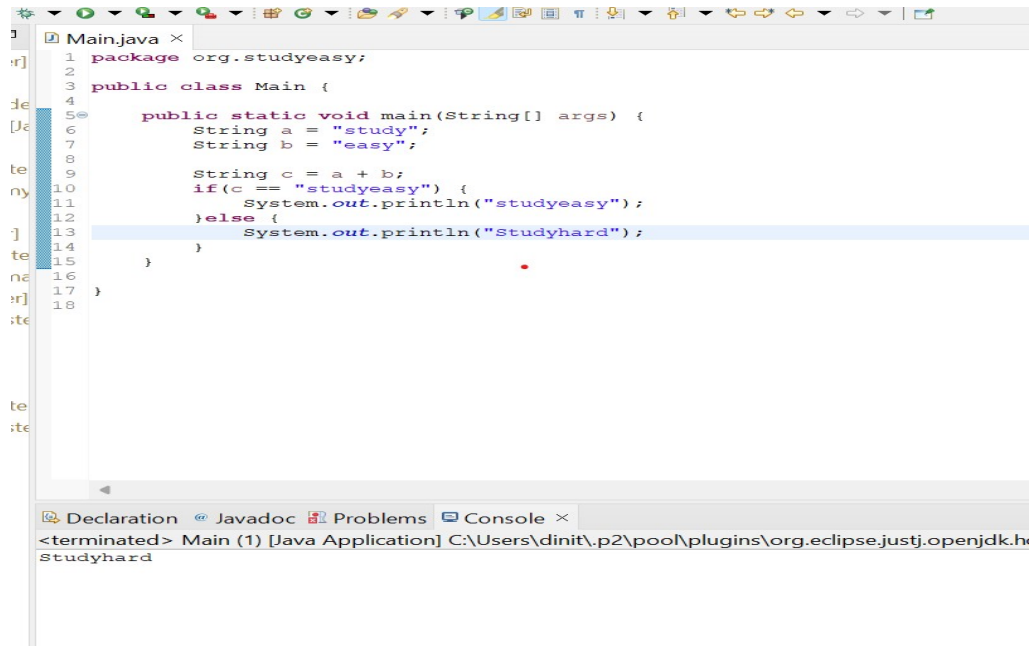


The screenshot shows the Eclipse IDE with a file named 'Main.java'. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String a = "study";
7         String b = "easy";
8
9         String c = a + b;
10        if(a == "study") {
11            System.out.println("Hi");
12        }else {
13            System.out.println("Hello");
14        }
15    }
16 }
17
18
```

The console output at the bottom reads: <terminated> Main (1) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Jan 29, 2022)

➤ Unexpected Output - String Example



The screenshot shows the Eclipse IDE with a Java file named Main.java. The code is as follows:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String a = "study";
7         String b = "easy";
8
9         String c = a + b;
10        if(c == "studyeasy") {
11            System.out.println("studyeasy");
12        }else {
13            System.out.println("Studyhard");
14        }
15    }
16 }
17
18
```

The console output at the bottom shows:

```
<terminated> Main (1) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v21-studyeasy
Studyhard
```

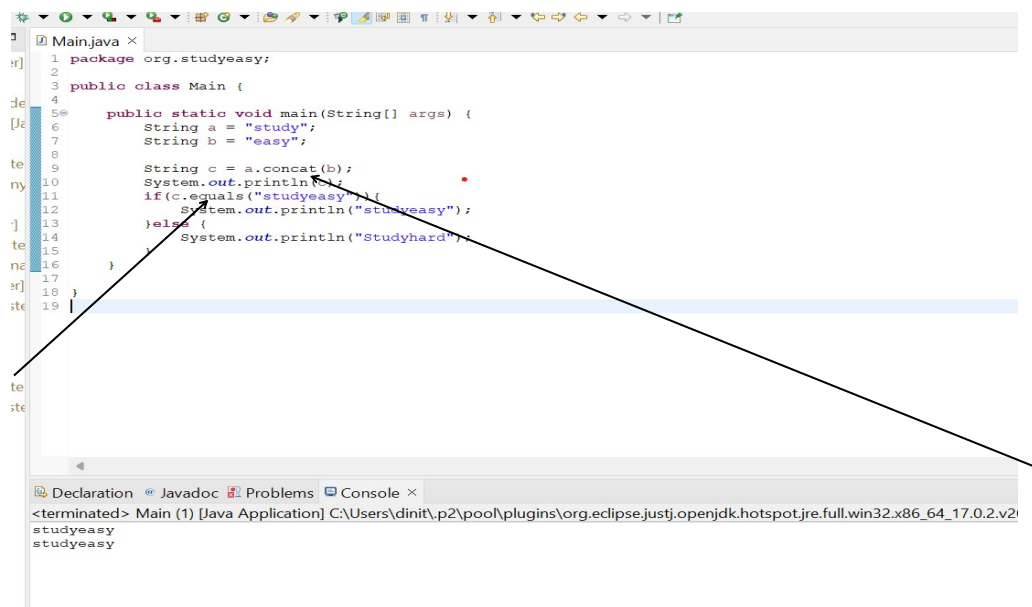
The output is incorrect because the string comparison uses the == operator, which compares object references instead of the string content.

➤ URL link for the Java String Class documentation

<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

➤ Solution for the above example

Can use String Manipulation Methods . Also refer to the above URL for more details.



The screenshot shows the Eclipse IDE with the same Java file, but the code has been corrected to use the equals() method for string comparison:

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         String a = "study";
7         String b = "easy";
8
9         String c = a.concat(b);
10        System.out.println(c);
11        if(c.equals("studyeasy")) {
12            System.out.println("studyeasy");
13        }else {
14            System.out.println("Studyhard");
15        }
16    }
17 }
18
```

The console output at the bottom now shows:

```
<terminated> Main (1) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v21-studyeasy
studyeasy
studyeasy
```

The output is now correct because the equals() method compares the string content.