

Credit Name: Class & Object-oriented development
Assignment Name: Digitextractor

Error Log Entry

What error message did you encounter (if any)?

I encountered an error message when I put && instead of + in the output area making the machine unable to understand what it has to input to the user apart from the text inside the quotation marks

What unexpected behavior did your program exhibit?

When I wanted to input any number it would provide red lines in the test area whenever I tried inputting an answer inside it because the code didn't know what was going on inside the brackets when it was outputting in the actual code.

What caused the issue? (e.g., syntax error, logic error, incorrect function usage, etc.)

The operator + is the correct way to join text and numbers in Java because it performs string concatenation, while the operator & & is only used to combine two boolean conditions. Since & & is meant strictly for logical operations, Java does not allow it to be used with strings or integers. When you try to use & & in a print statement, it causes a compile-time syntax error because the code does not follow Java's rules. The + operator works correctly because it automatically converts the number into a string and attaches it to the text. Without using +, Java has no valid way to combine the message and the variable, so the program cannot run.

Include a screenshot of specific lines of code.

```
}

    public void wholenumber(String letter) { // prints the whole number if "w" is entered
        if (letter.equals("w")) // checks if the input is "w"
            System.out.println("Whole number: " && number); // prints the whole number
    }

Enter a three-digit number:
767

Enter a letter:
(w) = whole number
(h) = hundreds place
(t) = tens place
(o) = ones place
(q) = quit
w
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The operator && is undefined for the argument type(s) String, int

    at mastery.Digitextractor.wholenumber(Digitextractor.java:21)
    at mastery.DigitextractorTest.main(DigitextractorTest.java:37)
```

How did you fix the issue?

I removed the "&&" and used the "+" symbol instead because that makes the machine realize that it should be added onto the output and it decides to add that on and that fixed the code completely and allowed it to work

Provide the corrected code or solution using a screenshot.

```
        number = n; // assigns the passed integer to the number variable
    }

    public void wholenumber(String letter) { // prints the whole number if "w" is entered
        if (letter.equals("w")) // checks if the input is "w"
            System.out.println("Whole number: " + number); // prints the whole number
    }
```

Enter a three-digit number:

432

Enter a letter:

(w) = whole number

(h) = hundreds place

(t) = tens place

(o) = ones place

(q) = quit

w

Whole number: 432

Enter a letter:

(w) = whole number

(h) = hundreds place

(t) = tens place

(o) = ones place

(q) = quit