

1. I made the program and use two different data types, which is double and big decimal.

Here's the summary of my observation:

- a. Big decimal commonly used in calculation programs for more precision when it comes to counting money.
- b. When I try calculating the result of 19.99 item price, 3 quantity and 0.11 tax rate using Double, the program set this as an output, which is inconvenient for calculating money. This result that is not precise can cause issues like money and law issues :

```
Enter the item price: 19.99
Enter the quantity: 3
Subtotal: $59.97
Tax (11%): $6.5967
Total: $66.5667
PS C:\Users\syifa\Documents\College\
```

- c. But when I try calculating using big decimal with the same price, quantity and tax rate, here's the result:

```
Enter the item price: 19.99
Enter the quantity: 3
Subtotal: $59.97
Tax (11%): $6.60
Total: $66.57
PS C:\Users\syifa\Documents\College\
```

2. Type Capacity Puzzle

- a. I'll use int, because the number of students is always a whole number and will never be a fraction.
- b. Char, because it is used to store a single character, such as the first letter of someone's name.
- c. Boolean, because a light switch only has two possible states, on(true) and off(false).
- d. Double, because distances between planets can be extremely large and may include decimals for precision.

3. Unexpected Results with Division

- a.  $7 / 2$  using int the result is 3 because both 7 and 2 are integers, so the result is also integer. In java, numbers after comma will be removed (from 3.5 to only 3).
- b.  $7 / 2$  use double, the result is 3.0 because the operation is still with integers, 7 and 2 remain integers,
- c. so the result remains 3 and this result is converted to double (3.0).

- d.  $7 / 2.0$  use double, and the result is 3.5 because in java, when the operand is double, the result will also be converted to double, so the number after comma finally get included (3.5).

#### 4. Literal Confusion

```
Exception in thread "main" java.lang.Error: Unresolved compilation problems:  
    The literal 3000000000 of type int is out of range  
    Type mismatch: cannot convert from double to float  
    Type mismatch: cannot convert from String to char
```

```
Long big = 3000000000;  
float f = 2.5;  
char c = "A";
```

- a. First, the 3000000000 mistaken by int because it doesn't have L at the end, the 3000000000 itself is an integer.
- b. 2.5 mistaken as double by default, when we want to use float, we can just add f at the end of the data.
- c. Quote mark is for string, if we want to use char just use apostrophy (').
- d. Here's the right code:

```
Long big = 3000000000L;  
float f = 2.5f;  
char c = 'A';
```

#### 5. Overflow Experiment

- a. This is the output:

```
-2147483648  
PS C:\Users\syifa
```

- b. The integer max value is **2,147,483,647**, so when we add 1, it becomes overflow and java wraps around like a clock to **-2147483648**.