Computer Science I (Java) – CSC 130 – Duke Hutchings

Notes (Day 4)

- Variables (a.k.a. "How math works in Java")
  - data types
  - type casting

How Math Works in Java

Opening Comment

Depending on how you do it, eight divided by three could be…

… 2

… 2.6666666666666665

… an error in your code's syntax

We must pay close attention to **data type** in Java programs.

Two Major Data Types

int	an integer value, and always an integer value
double	a decimal-point value

```
int x = 8;
int y = 3;
int z = x / y;

// z is now 2!
```

```
double d = 8;
double e = 3;
double f = d / e;

// f is now 2.6666666666666665!
```

Mixing Data Types

Java will figure out how to remain or increase in precision

Java demands that you tell it how to decrease in precision

```
int x = 8;
double d = 3;

double e = x / d; // OK with Java
int y = x / d;    // Java disallows
```

Typecasting: Telling Java to Decrease in Precision

When losing precision, you must explicitly indicate how.

This is called **type casting** (or just a “cast” for short).

```
int x = 8;
double d = 3;
// int y = x / d;       // Java disallows
int y = (int)(x / d); // Java allows: a “cast”

// note: y has the value 2, not 2.666...
```

Take Caution

Java stays "within" type until the last possible moment.

```
int x = 8;
int y = 3;
double d = x / y;

// d has the value 2.0, not 2.666...
```

Take Caution, Part 2

Cast one of the variables to force an “early” type change.

```
int x = 8;
int y = 3;
double d = x / (double)y;

// d now has the value 2.6666666666666665
```

Worksheet Time!

Let’s do some practice on the worksheet

We’ll also see how to do some code testing in Java