

CMSC131

Initial Values inside Classes

Initialization of **final** class variables

If a class field is marked as **final** then it needs to either be initialized where it is declared or be initialized by *each* of the constructors in the class so that by the time a new object is done being created, it has a value.

Initial values of *other class* variables

You can choose to explicitly initialize a class variable where it is declared in the class, even if it is not marked as **final**.

Explicit Initialization

file: Student.java

```
public class Student {  
    //Instance Variables  
    private int tokenLevel = 0;  
  
}
```

Initial values of *other class* variables

You can choose to explicitly initialize a class variable where it is declared in the class, even if it is not marked as **final**.

If you do not specify a default value for each class variable they will still obtain one when a new object of that class is created.

- Primitive numeric/byte types are initialized to 0.
- Primitive Boolean type is initialized to false.
- Primitive char type is initialized to the null character.
- Object references are initialized to null pointers.

Implicit Initialization

file: Student.java

```
public class Student {  
    //Instance Variables  
    private int tokenLevel = 0;  
    private String name; //set to null  
    private float tuition; //set to 0.0F  
  
}
```

NOTE/REMINDER

Local variables in Java are ***not*** initialized automatically.

This is true within ***any*** method (instance or static), not just the main method.

From a logical level, local variables **must** be assigned a value before being read.

This is actually checked at the compiler (or in the case of Eclipse, pre-compiler) level.

Copyright © 2012 : Evan Golub