

# Lesson 09: Eclipse and int Variables (W03D1)

Balboa High School

Michael Ferraro

August 26, 2015

- Mount your locker folder, start a terminal shell, and change to your locker's tmp directory from last class:  
`cd ~/MOUNTED/apcs-locker/tmp`
- Delete any HelloWorld files that may be in this directory:  
`rm HelloWorld.*`
- **From memory**, recreate HelloWorld. Remember the steps:
  - edit
  - compile
  - run

Students will continue their introduction to Eclipse for Java development and start using `int` variables in their programs.

- In math/science/engineering:
  - a placeholder for some as yet unknown value
  - represented by some symbol, e.g.,  $x$ ,  $\phi$ , and  $\alpha$

- In math/science/engineering:
  - a placeholder for some as yet unknown value
  - represented by some symbol, e.g.,  $x$ ,  $\phi$ , and  $\alpha$
- In computer science:
  - a reference to a value that is known and may change over time
  - represented by a letter (e.g., `a` & `i`) or string of letters (e.g., `age`, `typeofTicket`, & `myHeight`)

Example: Environment Variables in Windows

```
C:\> echo %PATH%
```

Example: Environment Variables in Windows

```
C:\> echo %PATH%
```

```
C:\Program Files\Java\JDK1.8.0_60\bin;
```

```
C:\Windows\System32;C:\;...
```

Example: Environment Variables in Windows

```
C:\> echo %PATH%
```

```
C:\Program Files\Java\JDK1.8.0_60\bin;
```

```
C:\Windows\System32;C:\;...
```

```
C:\> set PATH=C:\apcs
```



## Example: Environment Variables in Windows

```
C:\> echo %PATH%
```

```
C:\Program Files\Java\JDK1.8.0_60\bin;
```

```
C:\Windows\System32;C:\;...
```

```
C:\> set PATH=C:\apcs
```

```
C:\> echo %PATH%
```

## Example: Environment Variables in Windows

```
C:\> echo %PATH%
```

```
C:\Program Files\Java\JDK1.8.0_60\bin;
```

```
C:\Windows\System32;C:\;...
```

```
C:\> set PATH=C:\apcs
```

```
C:\> echo %PATH%
```

```
C:\apcs
```

# Variables

Example: Environment Variables in Windows

C:\> echo %PATH% ← string of characters referring to value in memory

C:\Program Files\Java\JDK1.8.0\_60\bin;

C:\Windows\System32;C:\;... ← initial state

C:\> set PATH=C:\apcs

C:\> echo %PATH%

C:\apcs ← mutated/alterd state

# New Project: FunWithInts

- Start Eclipse, point to `~/MOUNTED/apcs-locker/workspace0`
- Close the current project in Eclipse (watch me...)
- Create new project called `FunWithInts` and add new file called `IntegerFun.java`
- If you're waiting for classmates to catch up, take a few moments to recall (or look up) the mathematical definition of *integer*

- 1 Add the class declaration
- 2 Add the `main()` method

## IntegerFun.java: Class Definition

```
public class IntegerFun {  
  
}
```

## IntegerFun.java: main() Method

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
    }  
  
}
```

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
        ← add variable declarations for integers  
    }  
  
}
```



```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
        //declaring two integer vars  
        int a;  
        int b;  
  
    }  
  
}
```

---

Note: Java code may contain human-readable comments that will be ignored by the compiler that are there for the benefit of you and other developers. Such lines may begin with two forward slashes (//)

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
        //declaring two integer vars  
        int a;  
        int b;  
  
    }  
  
}
```

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
        //declaring two integer vars  
        int a;  
        int b;  
  
        System.out.println("alpha is " + a);  
        System.out.println("beta is " + b);  
    }  
  
}
```

---

Run the program.

# IntegerFun.java

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
        //declaring two integer vars  
        int a;  
        int b;  
  
        //set a and b equal to some values  
        a = 2;  
        b = 38;  
  
        System.out.println("alpha is " + a);  
        System.out.println("beta is " + b);  
    }  
}
```

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
        //declaring AND initializing integers  
        //simultaneously  
        int a = 2;  
        int b = 38;  
  
        System.out.println("alpha is " + a);  
        System.out.println("beta is " + b);  
    }  
  
}
```

# IntegerFun.java

```
public class IntegerFun {  
  
    public static void main(String[] args) {  
  
        //declaring AND initializing integers  
        //simultaneously  
        int a = 2;  
        int b = 38;  
        int c = a + b; //Java does math!  
  
        System.out.println("alpha is " + a);  
        System.out.println("beta is " + b);  
        System.out.println("they sum to " + c);  
    }  
  
}
```

# Now Experiment!

You have  $\approx 3$ min to...

- Try other math operations: -, \*, /
- Change the values stored in a & b
- After trying a few things, see what happens when you divide by zero

# Pushing Project to GitHub

- Starting with PS #1, you'll need to push code to GitHub.
- Let's push today's Eclipse project using `github-push.sh...`
- See our code at <http://github.com>.
- How do you make changes when at home?



# Quiz Next Class!

What questions do you have about the quiz?

---

- UNIX commands: `ls`, `cd`, `mkdir`, & `cp`
- Wildcards to specify multiple files
- Filesystem basics: root dir, special directory entries `.` and `..`
- Conversions: decimal  $\leftrightarrow$  binary  $\leftrightarrow$  hex
- Determining # of bits & bytes needed to represent decimal numbers
- Finding problems with a broken HelloWorld
- Carrying out the edit-compile-link process in Java from the command line
- Roles of an operating system

- Review lesson slides and exercises in preparation for next class' quiz.
- Log in to <http://github.com> and modify IntegerFun.java:
  - add a statement that prints this: I <3 APCS!
  - make sure to include a commit comment