

CS 113: Computer Science I

Instructor: Kelly Shiptoski

Week: 1, Lecture: 2

Lab & HW Update

We will finish the rest of the first lab next **Tuesday**.

HW #1 will be released ASAP and will be due next **Friday**.

(It will be very simple!)

Command Line Prompt

```
kshiptoski@comet:~$ javac Hi.java
```

All of the text that's on the left of where you are typing commands is called the **prompt**.

So here: `kshiptoski@comet:~$`

Vim Commands Refresher

`vim filename`: open a file in vim (`filename` = your file's name)

`i`: enter insert mode

`esc`: exit current mode

You can use the arrow keys to move around.

Navigating a Directory

`ls`: list files in current directory.

`cd`: change current directory.

`pwd`: print path to the current directory.

`mkdir`: make a directory.

Special Directories

<code>..</code>	Parent directory
<code>.</code>	Current working directory
<code>/</code>	Root directory
<code>/home/username</code>	Your home directory
<code>~</code>	Your home directory

Working with Paths

Two kinds of paths:

- Absolute paths
- Relative paths

Absolute: `/Users/kellyshiptoski/bryn_mawr`

Relative: `../../Users`

Directory Exercise (From Lab)

```
cs113/  
  labs/  
    lab1/  
  homeworks/  
    hw1/  
  demos/
```

Directory Exercise (From Lab)

```
cs113/  
  labs/  
    lab1/  
  homeworks/  
    hw1/  
  demos/
```

What is the relative path to:

- lab1/ from cs113/?
- demos/ from hw1/?
- cs113/ from hw1/?

Path Exercise

Visualize the directory structure from running the following commands.

```
> pwd
/home/kelly
> mkdir one
> cd one
> mkdir two
> touch talk.c
> cd ..
> touch listen.c
> cd
> touch sing.c
```

Note: **touch** is a command to create a new, empty file.

Simplest Java Program

```
class Hello {  
    public static void main(String[] args) {  
        System.out.println(x: "Hello World!");  
    }  
}
```

Compiling a Java Program

Converts java file (`.java`) to a file that the computer understands (`.class`)

```
javac filename.java → filename.class
```

Running a Java Program

```
java filename
```

(Don't include the *.class file extension).

But where do we actually call `java filename`?

Data Type Examples

`int` - whole numbers (ex: 7)

`double` - decimal numbers (ex: 1.25)

`String` - represents text (ex: "Kelly")

Variables

Variables can be created and changed by:

- Declaration → ex: `int x;`
- Assignment → ex: `x = 4;`
- Declaration & Assignment → ex: `int x = 4;`

Every **variable** has a **data type**.

Properties of Variables

- Name
- Data Type
- Location
 - Where on the computer is the variable stored?
 - More on this later...

Example: `String greeting;`

Properties of Variables

- Name
- Data Type
- Location
 - Where on the computer is the variable stored?
 - More on this later...

Example: `String greeting;`

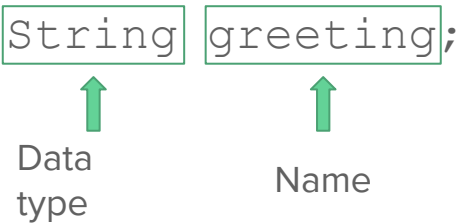


Data
type

Properties of Variables

- Name
- Data Type
- Location
 - Where on the computer is the variable stored?
 - More on this later...

Example: `String greeting;`



Data
type

Name

Printing Variables

```
class Hello {  
    public static void main(String[] args) {  
        int a = 1;  
        double b = 2.0;  
        String blah = "thisisastring";  
  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(blah);  
    }  
}
```

```
1  
2.0  
thisisastring
```

Printing Variables

```
class Hello {  
    public static void main(String[] args) {  
        int a = 1;  
        double b = 2.0;  
        String blah = "thisisastring";  
  
        System.out.println("a is " + a + ", b is " + b + ", and blah is " + blah);  
    }  
}
```

```
a is 1, b is 2.0, and blah is thisisastring
```

Variable Examples

a	b	c

Variable Examples

```
int a, b;
```

a	b	c

Variable Examples

```
int a, b;
```

a	b	c
undefined	undefined	-

Variable Examples

```
int a, b;
```

```
String c = "Kelly";
```

a	b	c
undefined	undefined	-

Variable Examples

```
int a, b;
```

```
String c = "Kelly";
```

a	b	c
undefined	undefined	-
undefined	undefined	"Kelly"

Variable Examples

```
int a, b;
```

```
String c = "Kelly";
```

```
a = 3;
```

a	b	c
undefined	undefined	-
undefined	undefined	"Kelly"

Variable Examples

```
int a, b;
```

```
String c = "Kelly";
```

```
a = 3;
```

a	b	c
undefined	undefined	-
undefined	undefined	"Kelly"
3	undefined	"Kelly"

Variable Examples

```
int a, b;  
String c = "Kelly";  
a = 3;  
b = a;
```

a	b	c
undefined	undefined	-
undefined	undefined	"Kelly"
3	undefined	"Kelly"

Variable Examples

```
int a, b;  
String c = "Kelly";  
a = 3;  
b = a;
```

a	b	c
undefined	undefined	-
undefined	undefined	"Kelly"
3	undefined	"Kelly"
3	3	"Kelly"

Rules for Naming Variables

Case sensitive.

Can't:

- Start with a number.
- Contain special characters (*, +, /, etc).
- No spaces.
- No special words (String, int, main, etc).

Converting Types (Numbers)

Double to integer:

- `(int) 3.14;`
- `int a = (int) 3.14; // Store the converted double in a var`

Storing an integer as a double:

- `double b = 6;`

Converting Types (Strings & Numbers)

Integer to String

- `int a = 23;`
- `String numMajors = String.valueOf(a);`

String to integer

- `int x = Integer.parseInt("40");`

String to double

- `double a = Double.parseDouble("40.11");`

Basic Operands

Operators: +, -, /, *, %...

Expressions:

- Example: 55 + c
- Operator is +
- Operands: **55**, **c**

Order of Operations

$24 + 10 / 2;$

$(24 + 10) / 2;$

Operations between floats and ints:

$1 / 3$

$1 / 3.0$

Exercise

Let's write a small Java program that converts **miles** to **kilometers** and **prints** out the result.

Kilometers = Miles x 1.6

Review

- How do you **print** in Java?
- What does a **declaration** statement do?
- What does an **assignment** statement do?
- Give me an example of an **illegal variable name**.
- Give me an example of an **operator**.