

Java basics

Objective

- Constants
- Variables
- Operations
- Primitive types
- Expressions
- Interactive programs

Computation

- Programmers frequently write small programs for computing useful things
- Example – body mass index (BMI)
 - Measure of fitness
 - Ratio of person’s weight to the square of the person’s height
 - Weight in kilograms, height in meters
 - Person of interest is 4.5 feet and weighs 75.5 pounds
- Metric conversions
 - Kilograms per pound 0.454
 - Meters per foot 0.3046

Common program elements

- Type
 - Set of values along with operators that can manipulate and create values from the set
- Primitive types support numeric, character, logical values
 - double and float
 - Values with decimals
 - byte, short, int, long
 - Integers
 - char
 - Characters (considered numeric)
 - boolean
 - Logical values
 - String, etc
- Basic operators
 - + addition
 - * multiplication
 - - subtraction
 - / division

Primitive Data Types

- **Numeric Data Types**
 - *Integer* numeric types are **8-bit byte**, **16-bit short**, **32-bit int**, and **64-bit long**.
 - The 8-bit byte data type in Java has replaced the old C and C++ char data type. Java places a different interpretation on the char data type, as discussed below.
 - There is no unsigned type specifier for integer data types in Java.
 - *Real* numeric types are **32-bit float** and **64-bit double**. Real numeric types and their arithmetic operations are as defined by the IEEE 754 specification.
 - A floating point *literal* value, like 23.79, is considered double by default; you must explicitly cast it to float if you wish to assign it to a float variable.

Primitive Data Types

- **Character Data Types**
 - Java language *character* data is a departure from traditional C. Java's **char** data type defines a **16-bit Unicode** character.
 - Unicode characters are unsigned 16-bit values that define character codes in the range 0 through 65,535. If you write a declaration such as `char myChar = 'Q'`; you get a Unicode (16-bit unsigned value) type initialized to the Unicode value of the character Q.
 - By adopting the Unicode character set standard for its character data type, Java language applications are amenable to internationalization and localization, greatly expanding the market for world-wide applications.

Primitive Data Types

- **Boolean Data Types**
 - Java added a Boolean data type as a primitive type, tacitly ratifying existing C and C++ programming practice, where developers define keywords for TRUE and FALSE or YES and NO or similar constructs.
 - A Java boolean variable assumes the value true or false. A Java programming language boolean is a distinct data type; unlike common C practice, a Java programming language boolean type can't be converted to any numeric type.

Primitive Data Types

- **Arrays**
 - In contrast to C and C++, the Java programming language *arrays* are first-class language objects. An array in the Java programming language is a real object with a run-time representation. You can declare and allocate arrays of any type, and you can allocate arrays of arrays to obtain multi-dimensional arrays. You declare an array of, say, Points (a class you've declared elsewhere) with a declaration like this:
`Point myPoints[];`

Primitive Data Types

- **Strings**
 - *Strings* are Java programming language objects, not pseudo-arrays of characters as in C.
 - There are actually two kinds of string objects: the `String` class is for read-only (immutable) objects. The `StringBuffer` class is for string objects you wish to modify (mutable string objects). Although strings are Java programming language objects, Java compiler follows the C tradition of providing a syntactic convenience that C programmers have enjoyed with C-style strings, namely, the Java compiler understands that a string of characters enclosed in double quote signs is to be instantiated as a `String` object. Thus, the declaration:
`String hello = "Hello world!";`
instantiates an object of the `String` class behind the scenes and initializes it with a character string containing the Unicode character representation of "Hello world!". Java technology has extended the meaning of the `+` operator to indicate *string concatenation*.

Common program elements

- **Constant**
 - Symbolic name for memory location whose value does not change
 - `KILOGRAMS_PER_POUND`
- **Variable**
 - Symbolic name for memory location whose value can change
 - `weightInPounds`

Program outline for BMI.java

// Purpose: Compute BMI for given weight and height

```
public class BMI {  
  
    // main(): application entry point  
    public static void main(String[] args) {  
        // define constants  
  
        // set up person's characteristics  
  
        // convert to metric equivalents  
  
        // perform bmi calculation  
  
        // display result  
    }  
}
```

```
public static void main(String[] args) {  
    // define constants  
    final double KILOGRAMS_PER_POUND = 0.454;  
    final double METERS_PER_FOOT = 0.3046;  
  
    // set up person's characteristics  
    double weightInPounds = 75.5; // our person's weight  
    double heightInFeet = 4.5; // our person's height  
  
    // convert to metric equivalents  
    double metricWeight = weightInPounds *  
        KILOGRAMS_PER_POUND;  
    double metricHeight = heightInFeet * METERS_PER_FOOT;  
  
    // perform bmi calculation  
    double bmi = metricWeight / (metricHeight * metricHeight);  
  
    // display result  
    System.out.println("A person with");  
    System.out.println("  weight " + weightInPounds + " lbs");  
    System.out.println("  height " + heightInFeet + " feet");  
    System.out.println("has a BMI of " + Math.round(bmi));  
}
```

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

KILOGRAMS_PER_POUND 0.454

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

METERS_PER_FOOT 0.3046

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

weightInPounds 75.5

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

heightInFeet 4.5

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

metricWeight 34.2770

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

metricHeight 1.3706

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

bmi 18.2439

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

Operator evaluation depend upon its operands

```

public static void main(String[] args) {
    // define constants
    final double KILOGRAMS_PER_POUND = 0.454;
    final double METERS_PER_FOOT = 0.3046;

    // set up person's characteristics
    double weightInPounds = 75.5; // our person's weight
    double heightInFeet = 4.5; // our person's height

    // convert to metric equivalents
    double metricWeight = weightInPounds *
        KILOGRAMS_PER_POUND;
    double metricHeight = heightInFeet * METERS_PER_FOOT;

    // perform bmi calculation
    double bmi = metricWeight / (metricHeight * metricHeight);

    // display result
    System.out.println("A person with");
    System.out.println(" weight " + weightInPounds + " lbs");
    System.out.println(" height " + heightInFeet + " feet");
    System.out.println("has a BMI of " + Math.round(bmi));
}

```

```

// Purpose: Demonstrate char arithmetic

public class LowerToUpper {

    // main(): application entry point
    public static void main(String[] args) {
        // set lower case character of interest
        char lowerCaseLetter = 'c';

        // convert to uppercase equivalent
        int upperCaseLetter = 'A' + (lowerCaseLetter - 'a');

        // display result
        System.out.println("Uppercase equivalent of");
        System.out.println(" " + lowerCaseLetter);
        System.out.println("is");
        System.out.println(" " + (char)upperCaseLetter);
    }
}

```

Expressions

- What is the value used to initialize expression
`int expression = 4 + 2 * 5;`
- What value is displayed
`System.out.println(5 / 2.0);`
- Java rules in a nutshell
 - Each operator has a **precedence** level and an **associativity**
 - Operators with higher precedence are done first
 - * and / have higher precedence than + and -
 - Associativity indicates how to handle ties
 - When floating-point is used the result is floating point

Question

- Does the following statement compute the average of double variables a, b, and c? Why

```
double average = a + b + c / 3.0;
```

Interactive programs

- Programs that interact with their users through statements performing input and output
- BMI.java
 - Not interactive – weight and height are fixed

Interactive program for BMI

```
// Purpose: Compute BMI for user-specified weight and height

import java.io.*;

public class BMIcalculator {

    // main(): application entry point
    public static void main(String[] args) throws IOException {
        // defining constants
        final double KILOGRAMS_PER_POUND = 0.454;
        final double METERS_PER FOOT = 0.3046;

        // displaying legend
        System.out.println("BMI Calculator\n");

        // set up input stream
        BufferedReader stdin = new BufferedReader(
            new InputStreamReader(System.in));

        // get person's characteristics
        System.out.print("Enter weight (lbs): ");
        double weight = Double.parseDouble(stdin.readLine());

        System.out.print("Enter height (feet): ");
        double height = Double.parseDouble(stdin.readLine());
    }
}
```

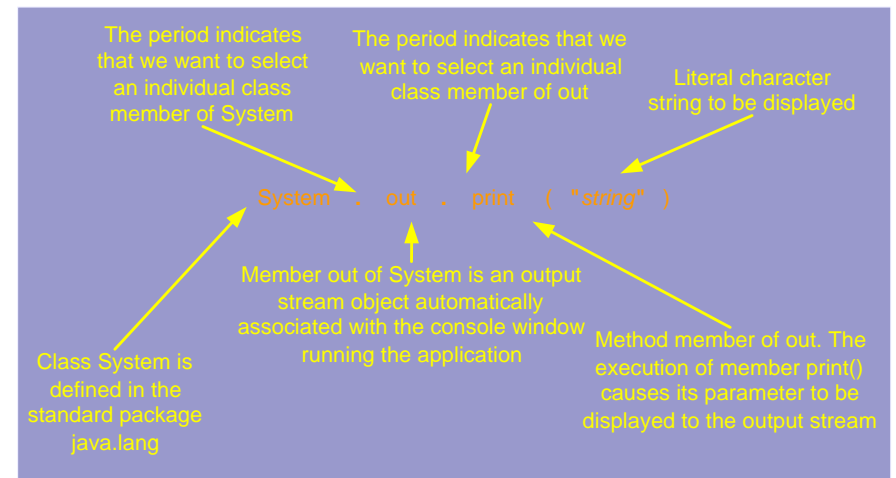
Interactive program for BMI

```
// convert to metric equivalents
double metricWeight = weight * KILOGRAMS_PER_POUND;
double metricHeight = height * METERS_PER FOOT;

// perform bmi calculation
double bmi = metricWeight / (metricHeight * metricHeight);

// display result
System.out.println("A person with");
System.out.println("  weight " + weight + " (lbs)");
System.out.println("  height " + height + " (feet)");
System.out.println("has a BMI of " + bmi);
}
}
```

Selection



Support for interactive console programs

`import java.io.*;`

- An import statement indicates that the program required access to other Java recourses
- In creating and accessing the input stream the input stream object for getting input values, the program uses classes that are defined in standard Java software library java.io
- A software library is known as a package or application programming interface (API)
- JAVA API Can be found in <http://java.sun.com/j2se/1.4.2/docs/api/>
- Only java.lang is implicitly to a program, it is necessary to import the needed classes from the io API
- The * indicates all resources of java.io are to be made available to the program, it is called the wildcard character
- Or we can use

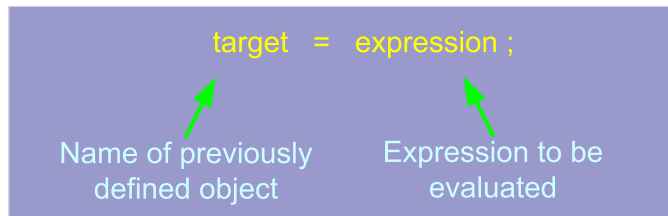
```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
```

```
BufferedReader stdin = new BufferedReader(
    new InputStreamReader(System.in));
// get person's characteristics
System.out.print("Enter weight (lbs): ");
double weight = Double.parseDouble(stdin.readLine());
```

- Initializes a variable `stdin` of type `BufferedReader`
- `stdin` then can access the a `BufferedReader` object, which is an input stream
- `BufferedReader` objects have a method `readLine()` that can extract character strings from an input stream (by default, `System.in` is associated with the keyboard)
- `BufferedReader` has a variable which is a reference type variable
- The new expression constructs a new `BufferedReader` object
- `readline()` is a method of `BufferedReader` object
- `parseDouble` is a method of `Double` object

Primitive variable assignment

- Assignment operator =
 - Allows the memory location for a variable to be updated

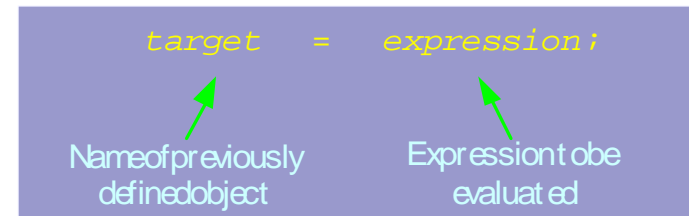


- Consider
`int j = 11;`
`j = 1985;`

j 11

Primitive variable assignment

- Assignment operator =
 - Allows the memory location for a variable to be updated



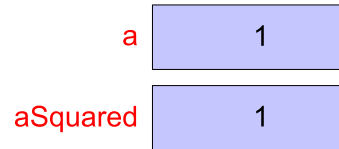
- Consider
`int j = 11;`
`j = 1985;`

j 1985

Primitive variable assignment

- Consider

```
int a = 1;  
int aSquared = a * a;  
a = 5;  
aSquared = a * a;
```



- Consider

```
int i = 0;  
i = i + 1;
```

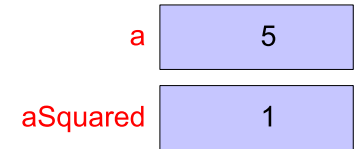
- Consider

```
int asaRating;  
asaRating = 400;
```

Primitive variable assignment

- Consider

```
int a = 1;  
int aSquared = a * a;  
a = 5;  
aSquared = a * a;
```



- Consider

```
int i = 0;  
i = i + 1;
```

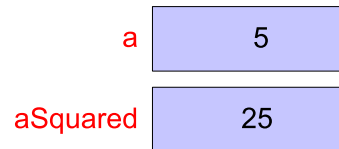
- Consider

```
int asaRating;  
asaRating = 400;
```

Primitive variable assignment

- Consider

```
int a = 1;  
int aSquared = a * a;  
a = 5;  
aSquared = a * a;
```



- Consider

```
int i = 0;  
i = i + 1;
```

- Consider

```
int asaRating;  
asaRating = 400;
```

Primitive variable assignment

- Consider

```
int a = 1;  
int aSquared = a * a;  
a = 5;  
aSquared = a * a;
```



- Consider

```
int i = 0;  
i = i + 1;
```

- Consider

```
int asaRating;  
asaRating = 400;
```

Primitive variable assignment

- Consider

```
int a = 1;
int aSquared = a * a;
a = 5;
aSquared = a * a;
```
- Consider

```
int i = 0;
i = i + 1;
```
- Consider

```
int asaRating;
asaRating = 400;
```

i 1

Primitive variable assignment

- Consider

```
int a = 1;
int aSquared = a * a;
a = 5;
aSquared = a * a;
```
- Consider

```
int i = 0;
i = i + 1;
```
- Consider

```
int asaRating;
asaRating = 400;
```

asaRating -

Primitive variable assignment

- Consider

```
int a = 1;
int aSquared = a * a;
a = 5;
aSquared = a * a;
```
- Consider

```
int i = 0;
i = i + 1;
```
- Consider

```
int asaRating;
asaRating = 400;
```

asaRating 400

Primitive variable assignment

- Consider

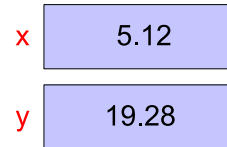
```
double x = 5.12;
double y = 19.28;
double rememberX = x;
x = y;
y = rememberX;
```

x 5.12

Primitive variable assignment

- Consider

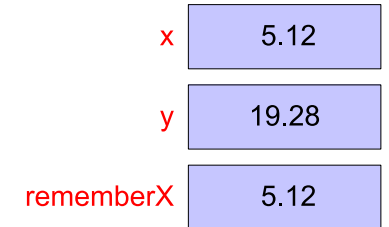
```
double x = 5.12;  
double y = 19.28;  
double rememberX = x;  
x = y;  
y = rememberX;
```



Primitive variable assignment

- Consider

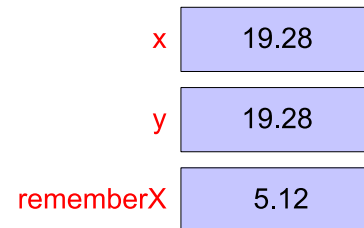
```
double x = 5.12;  
double y = 19.28;  
double rememberX = x;  
x = y;  
y = rememberX;
```



Primitive variable assignment

- Consider

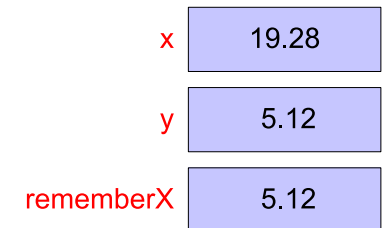
```
double x = 5.12;  
double y = 19.28;  
double rememberX = x;  
x = y;  
y = rememberX;
```



Primitive variable assignment

- Consider

```
double x = 5.12;  
double y = 19.28;  
double rememberX = x;  
x = y;  
y = rememberX;
```



Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

- Consider

```
int i = 4;
++i;
System.out.println(i);
System.out.print(++i);
System.out.println(i++);
System.out.println(i);
```

Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

i 4

- Consider

```
int i = 4;           // define
++i;
System.out.println(i);
System.out.print(++i);
System.out.println(i++);
System.out.println(i);
```

Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

i 5

- Consider

```
int i = 4;
++i;           // increment
System.out.println(i);
System.out.print(++i);
System.out.println(i++);
System.out.println(i);
```

Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

i 5

- Consider

```
int i = 4;
++i;
System.out.println(i); // display
System.out.print(++i);
System.out.println(i++);
System.out.println(i);
```

Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

i 

- Consider

```
int i = 4;
++i;
System.out.println(i);
System.out.print(++i); // update then display,
    i.e., do i=i+1, then output new i
System.out.println(i++);
System.out.println(i);
```

Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

i 

- Consider

```
int i = 4;
++i;
System.out.println(i);
System.out.print(++i);
System.out.println(i++); // display then update,
    i.e., output i first, then do i=i+1
System.out.println(i);
```

Increment and decrement operators

- ++
 - Increments a number variable by 1
- --
 - Decrements a numeric variable by 1

i 

- Consider

```
int i = 4;
++i;
System.out.println(i);
System.out.print(++i);
System.out.println(i++);
System.out.println(i); // display
```

Exercise

- What are the answers by using Java
 - 5/4
 - 5.0/4
 - 100/2*5
- What is the output of the following code
 - System.out.println(1+1+"1");
- How about this?
 - System.out.println(1+1+'1');
- What are the values of a, b and c?

```
int a = 1;
int b = a++;
int c = ++b;
```

Assignment (Due: Jan 30, Tuesday before
noon)

- 2.21
- 2.29
- 2.36
- 2.39
- 2.40