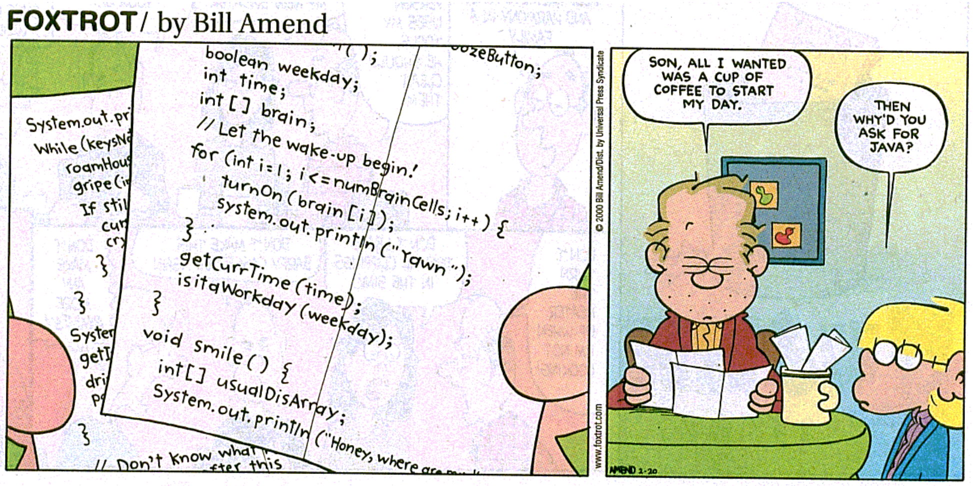
Mr. Fahrenbacher
AP Computer Science
Unit 2 - Data

Name: ______________________________

List of Java Concepts and Vocabulary From This Unit

- primitive data type
- Variable declaration
- Variable naming
- Variable assignment
- Variable initialization
- Block (class, method)
- Style conventions
- Order of operations (precedence)
- Remainder operator
- Integer division
- Output statement
- Casting
- Compound operator
- String concatenation
- Escape Sequences
- Three types of errors
- Constant
- Math functions and constants
- Randomness
- Comments
- User Input
- Arrays


FOXTROT / by Bill Amend
© 2000 Bill Amend/Dist. by Universal Press Syndicate
www.foxtrot.com
SON, ALL I WANTED WAS A CUP OF COFFEE TO START MY DAY.
THEN WHY'D YOU ASK FOR JAVA?
AMEND 1-20

```
System.out.pr
While (keysN
   roamHous
   gripe (i
   If stil
      cur
      cr
   }
}
Syster
getI
dri
P
}
// Don't know what
   fter this
```

```
          ( );                    ozeButton;
boolean weekday;
int time;
int [] brain;
// Let the wake-up begin!
for (int i=1; i<=numBrainCells; i++) {
   turnOn (brain[i]);
   System.out.println ("Yawn");
}
getCurrTime (time);
isitaWorkday (weekday);
}
void Smile () {
   int[] usualDisArray;
   System.out.println ("Honey, where are m
```

Primitive Data and Strings

	Code	Notes

Declaring (introducing) variables
type varName;
	`int coins;`
	`double height;`
	`String name;`
		(32 bits => 2^32 = 4 billion) [for integers]
		-2 billion < int < 2 billion (rough)
		64 bits (decimal values - roughly 15 decimal places))
		Max around 10^308
		holds text

Assigning (given value to) variables
varName = data;
	`coins = 5;`
	`height = 5.0;`
	`name = "Mario";`	

Declaring and Assigning variables
type varName = data;
	`int playerNumber = 1;`
	`double speed = 3.5;`
	`String world = "1-1";`	

Multiple declarations and/or assignments	`int goombas = 2, stars = 1, mushrooms;`	

The very first time a variable is assigned a value is called an initialization statement.

An Example Program

Read the following program and identify the parts you understand. Use vocabulary we developed on the previous page - by precise! Try to identify what other parts of the program are doing that look new.

```
public class TipCalculator {

      public static void main(String[] args) {

            double bill = 45.95;           //declaration of bill as a decimal, and initializes to 45.95
            double tipPercent;             //declares tipPercent as a decimal
            tipPercent = 0.15;             //initializes tipPercent

            double tipAmount = 0;          //declaration of tipAmount as a decimal, and initializes to 0
            tipAmount = bill * tipPercent;       //assign tipAmount the result of multiplyinh 45.95 and 0.15
            String message = "The tip amount is $";    //declares a string variable and initializes it

            System.out.print(message);
            System.out.print(tipAmount);
      }

}
```

Output when this program is run: **The tip amount is $6.8925**

Camel Case => variableNamesShouldStartWithLowerCaseAndCapitalizeEveryWordAfter

Comments

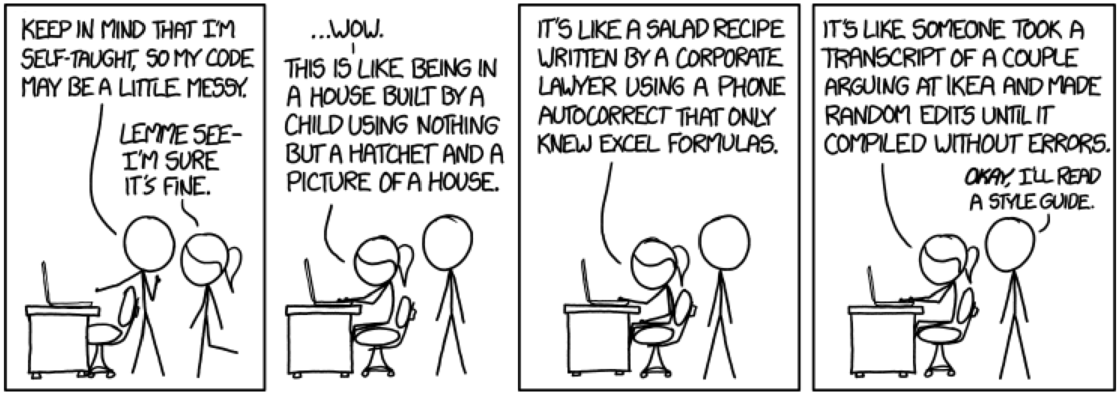
Read the information below about comments in the example program (the bolded parts are the comments)

```
/*      Name: Mr. Fahrenbacher
        Date: 8/10/2017
        Description: Comments are sections of code that are the documentation of the program - they do not impact
        the running of the program. At the top of a program is a good place to put your name and a general
        description of your program
*/

public class Area { //This class will be used for calculating the area of a square

        /* Below is the main method */
        public static void main(String[] args) {

                //the length of each side of a square
                int side = 5;

                /* Area of a square is the product of the sides */
                int area = side * side;

                //Comments are useful for turning code "off" for testing purposes, like the line below
                //System.out.println(area);
        }
}
```

Style

[The 17 Bits of Style]

Bit 0, 10, and 11: Comment your code
Bit 2: Use new lines to create paragraphs
Bit 3: Each statement goes on one line
Bit 4: Watch your braces
Bit 5: Use spacing
Bit 6: Remove extra paranthesis
Bit 8 and 9: Don't over comment obvious code
Bit 12: Use meaningful variable names
Bit 14: Use camel case for variable names
Bit 16: Write code knowing someone else will need to read it


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE— I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

A couple of math questions

1) 5 - 3 + 2 = 4

2) $6 \div 2 \cdot (5 + 3 - 4)$= 12

________3) 100 students are going to form baseball teams, with each team having 9 players. Will any student **not** be on a team? If so, how many?
________remainder = 1

Operations on int's and double's.

Precedence	Evaluate First		Evaluate Second (left to right)			Evaluate Last (left to right)	
Name	**Parentheses**	**Exponent**	**Multiply**	**Divide**	**Remainder**	**Add**	**Subtract**
Symbol	( )	There is no symbol	*	/	%	+	-

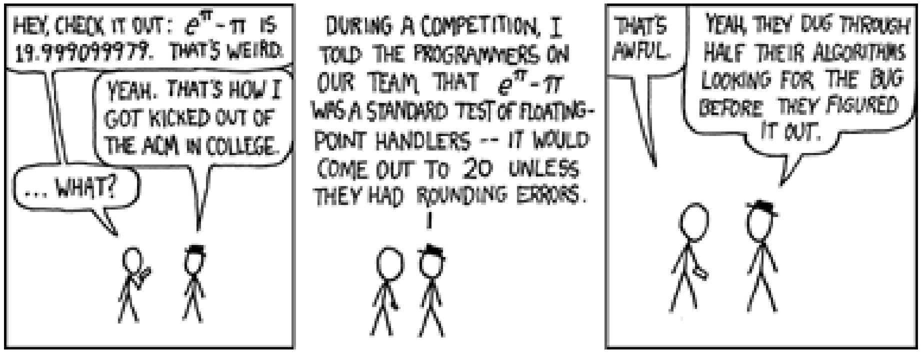
Doubles Are Imprecise

```
public class SimpleSum {

      public static void main(String[] args) {
            System.out.println(0.1 + 0.1 + 0.1);
            System.out.println(0.1 / 0.2);
            System.out.println(4.1 / 12.3);
      }
}
```

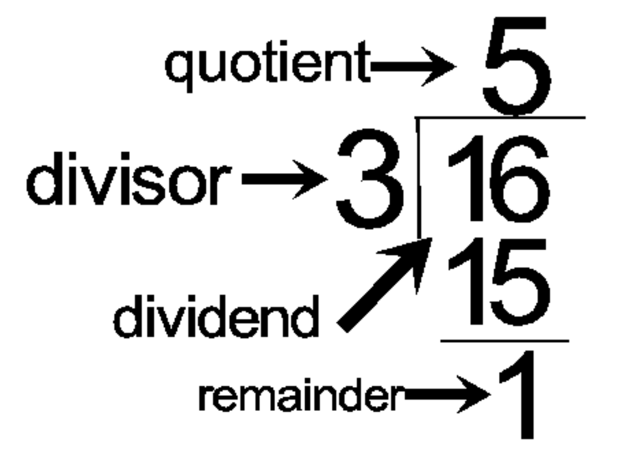
Output:

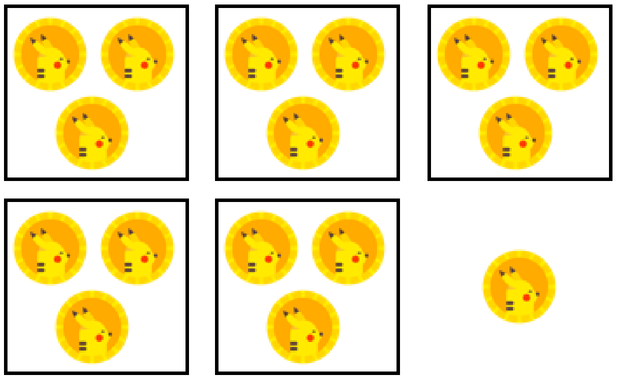
```
0.30000000000000004
0.5
0.33333333333333326
```

Why? Storing decimal values in binary is complicated.


HEY, CHECK IT OUT: $e^{\pi} - \pi$ IS 19.999099979. THAT'S WEIRD.
YEAH. THAT'S HOW I GOT KICKED OUT OF THE ACM IN COLLEGE.
... WHAT?
DURING A COMPETITION, I TOLD THE PROGRAMMERS ON OUR TEAM THAT $e^{\pi} - \pi$ WAS A STANDARD TEST OF FLOATING-POINT HANDLERS -- IT WOULD COME OUT TO 20 UNLESS THEY HAD ROUNDING ERRORS.
THAT'S AWFUL.
YEAH, THEY DUG THROUGH HALF THEIR ALGORITHMS LOOKING FOR THE BUG BEFORE THEY FIGURED IT OUT.

Remainder and Division (Quotient) Operator (for integers)





16 % 3 = 1

16 / 3 = 5

Verbal Model	Code Model	Answer
A class of 20 students is going to break into groups of size three. How many students will not be in a group of three?	int leftOver = 20 % 3;	2
A class of 20 students is going to break into groups of size three. How many full groups will there be?	int fullGroups = 20 / 3;	6

Evaluate the expressions to the right. Use the Expression Evaluator to see if your answers are correct!	18 % 7 = 4	4 % 10 = 4	24 % 6 = 0	1453 % 10 = 3
	18 / 7 = 2	4 / 10 = 0	24 / 6 = 4	1453 / 10 = 145

Notice in the second column what happens when you divid a small integer by a big integer! This is a common bug.

The last column is important! It's a technique for determing the ones place and the rest of the number!

When dividing two decimal values, the result is similar to what you would get on a calcaulot (1.0 / 5.0 = 0.2)

Another Example Program / Tracing

An important part about understanding code is called *tracing*. We'll build a trace table to understand this code.

```
public class TimeConverter {

      public static void main(String[] args) {

            int totalSeconds = 10000;
            int seconds, minutes, hours;

            hours = totalSeconds / 3600;
            totalSeconds = totalSeconds % 3600;       //update operation

            minutes = totalSeconds / 60;
            totalSeconds = totalSeconds % 60;

            seconds = totalSeconds;

            System.out.println("The hours are:  " + hours);
            System.out.println("The minutes are:  " + minutes);
            System.out.println("The seconds are:  " + seconds);
      }
}
```

Output when this program is run:

The hours are: 2
The minutes are: 46
The seconds are: 40

tS	s	m	h
~~10000~~	40	46	2
2800			
40			

Statements Are NOT Relationships

```
public class TaxingSituation {

      public static void main(String[] args) {
            double bill = 35.0;
            double taxRate = 0.1;
            double tax = taxRate * bill;   //tax: 3.5

            bill = bill - 10; //we had a coupon!   bill: 25.0

            System.out.println("Total tax: " + tax);
      }
}
```

Output when this program is run: **Total Tax: 3.5**
`//Did you think it should be 2.5?`

Mixed Type Operations (int / double)

Below are some segments of code. The left represents valid syntax, while the right side is invalid syntax. Explain what you think is invalid about the code on the right, then also list out what values you think the variables a through i have and why.

```
int a, d, e, x, y, z;
double b, c, f, g, h, i;
```

Legal Syntax		Illegal Syntax (Error)	
Code	Calculated Values and Why	Code	Why

a = 3;
b = 3.0;
c = 3;
	a = __3__
	b = __3.0__
	c = _3.0___
		x = 3.0;	Because x is an int, it can only hold numbers without a decimal

d = (int)5.4;	d = __5__	y = (double)5;	Because y is an int, it can't hold decimal data

e = 5/4;
f = 5/4.0;
	e = __1__
	f = __1.25__
		z = 5/4.0;	

g = 5/4;
h = (double)(5/4);
i = (double)5/4;
	g = _1.0___
	h = _1.0_
	i = _1.25___		

Cast Operator (int ⇔ double)

```
double temperature = -8.95;
int whole = (int)temperature;
                //-8
```

```
int sum = 2 + 3 + 4 + 5;
double average = (double)sum / 4;
                //3.5
```

Compound Operators (int, double)

```
int x = 10;
```

Regular Update Operation	`x = x + 5;`	`x = x - 4;`	`x = x * 5;`	`x = x / 6;`	`x = x % 7;`	`x = x + 1;`	`x = x - 1;`
Compound Operation	x += 5;	x -= 4;	x *= 5;	x /= 6;	x %= 7;	x++;	x--;

Age Trick: Link

Trace the program below. What is the output?

```
public class NumberTrick {

        public static void main(String[] args) {

                int age = 36;
                age *= 10;

                int num = 5; //could be anything between 1 and 9
                num *= 9;

                int result = age - num;

                int ones = result % 10;
                int rest = result / 10;

                int sum = ones + rest;
                System.out.println(sum);
        }
}
```

Phone Number Trick: Link

Trace the program below. What is the output?

```
public class PhoneTrick {

        public static void main(String[] args) {

                int first = 647;
                int second = 7744;

                first = first * 80;
                first++;
                first *= 250;

                first += second;
                first = first + second;

                first -= 250;
                first /= 2;

                System.out.println(first);
        }
}
```

Constants

A new keyword is introduced in the code below. Identify the keyword, then explain what purpose the keyword serves (compare the legal and illegal code).

Legal Syntax	Illegal Syntax

`double circumference = 0;`
`final` `double pi = 3.14;`
`double r = 5;`
`circumference = 2 *pi * r;`
	`final` `double circumference = 0;`
	`double pi = 3.14;`
	`double r = 5;`
	`circumference = 2 * pi * r;`

Math Functions

Java has many built in math functions. Several are listed in the table below. Identify what you think each command does.

```
int x;
double y;
```

Command	variable's value	What the command does
`x = Math.abs(-5);`	5	absolute value
`x = Math.max(2, 7);`	7	pickest the biggest from 2 values
`x = Math.min(2, 7);`	2	pickest the smallest from 2 values
`y = Math.pow(3, 4);`	81.0	exponent function!
`y = Math.sqrt(25);`	5.0	square root!

A common error is to think that Math.min(2, 3, 4) is valid syntax, but the min function only supports two inputs. Try to think of another way to determine the smallest value between 3 values using the min function.

```
int x = Math.min(Math.min(2, 3), 4);
```

Note: `Math.abs()`, `Math.max()`, and `Math.min()` will give a double result if at least one of the inputs is a double. `Math.pow()` and `Math.sqrt()` always give a double result.

Randomness

Below is some new code that allows you to generate one random number that represents a class in high school (9th-12th). Identify the different commands that generated the random numbers, and explain why you think they are being used. For example, what do you think would happen if you changed the numbers?

```
public class ContestWinner {

      public static void main(String[] args) {

            int winningClass = (int)(Math.random() * 4) + 9;       //4 -> number of grades, 9 => first grade
            System.out.print("The winning class is the " + winningClass + "th graders!");
      }

}
```

Example output when this program is run: **The winning class is the 12th graders!**

Randomness Analyzed

```
int x, double y;
```

Command	Stored Value (Math Notation)	Stored Value (Examples)
`y = Math.random();`	**[0, 1)**	0, 0.25, 0.16845734, 0.9999999999998
`x = (int)(Math.random() * 4);`	**(int)[0, 4) =>{0, 1, 2, 3}**	3, 3, 3, 3, 3, 3, 1, 2, 3, 1, 3, 0, 3, 2
`x = (int)(Math.random() * 4) + 9;`	{9, 10, 11, 12}	
`x = (int)(Math.random() * nums) + start;`	{start, …, nums+start-1}	

Randomness Mistakes

Command	Desired Values	Actual Values	Why
`x = (int)Math.random()*6 + 1;`	1, 2, ..., 6	1	Missing Paranthesis
`x = (int)(Math.random()*10) + 2;`	2, 3, ..., 12	2, 3, …, 11	They need to multiply by 11 to get 11 numbers!
`x = (int)(Math.random()*10);`	1, 2, ..., 10	0, …, 9	They needed to add by the start number! (+1)
`x = (int)(Math.random()*3) + 2;`	3, 4	2, 3, 4	Starting number and number of numbers is switched!

Randomness Questions

1) How would you store a random value between 2 and 12 in a variable named sum?

int sum = (int)(Math.random()*11) + 2;

2) Would your answer need to #1 need to change if you were generating the sum of two dice rolls?

_____int roll1 = (int)(Math.random()*6) + 1;
int roll2 = (int)(Math.random()*6) + 1;
int sum = roll1 + roll2;

Math Constants

These two constants are provided by the Math class. The first you know, the second you may not (if you don't, no worries - we won't really use it).

Constant	Value
`Math.PI`	3.141592653589793
`Math.E`	2.718281828459045

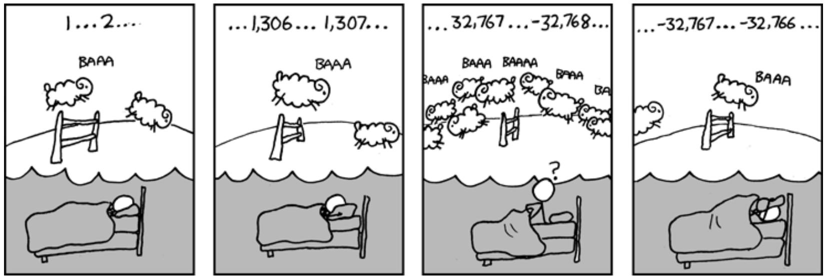
Example use:

```
double circleArea = Math.PI * Math.pow(5.0, 2);            //calculate the area of a circle
double loan = 1000 * Math.pow(Math.E, 0.05 * 10); //calculate the value of a loan, compounding continuously
```

Integer constants

In addition to the Math class, there is another class called the Integer class that has two constants that can be useful.

Constant	Value
`Integer.MAX_VALUE`	2147483647
`Integer.MIN_VALUE`	-2147483648



Example use:

```
int value = Integer.MAX_VALUE;
value++;
System.out.println(value);            //what do you think is printed?
```

String Concatenation

Strings also support being added together (and being add with other types!)

```
    String x;
    int age = 36;
    int height = 6;
```

Operation	Value in x
`x = "I am " + age + " years old";`	"I am 36 years old"
`x = "I am " + age + " years old and " + height + " feet tall";`	"I am 36 years old and 6 feet tall"
`x = "I am 3" + height + " years old";`	**"I am 36 years old"**

Explain why you think the results below are occurring.

Operation	Value in x	Why
`x = "My stats are " + age + height + "!";`	"My stats are 366!"	Order of operations!
`x = "My stats are " + (age + height) + "!";`	"My stats are 42!"	Order of operations!

Identify the mistakes in the code below.

Operation	Error and How to Fix
x = "My age is " **+** age;	
`x = "I am " + age + " years old and " + height + " feet tall";`	

Escape Sequences

Read the code below and compare it with the output. Identify the special pairs of symbols that are formatting the output in interesting ways.

```
public class MessageInABottle {

      public static void main(String[] args) {

            String msgA = "August 10th, 1815\n\nTo whom it may concern\n\n";
            String msgB = "\tThanks for finding my \"message\"!  I've been stranded on St. Helena for many"; String
            msgC = " days.\n\tPlease send help!\n\n";
            String msgD = "Sincerely,\n\\Napoleon\\";

            System.out.print(msgA + msgB + msgC + msgD);
      }
}
```

Output:

```
August 10th, 1815

To whom it may concern

      Thanks for finding my "message"!  I've been stranded on St. Helena for many days.
      Please send help!

Sincerely,
\Napoleon\
```

Types of Errors

The program below has three errors, each of a different kind. Find the errors.

```
public class WorkingTheNumbers {

      public static void main(String[] args) {          //missing type declaration is a SYNTAX error

            int linesOfCode = 2147483647;
            int people = 0;

            linesOfCode++;                              //making a logical mistake is a LOGIC error

            int rate = linesOfCode / people;            //dividing by 0 is a RUNTIME error
            System.out.println("The rate at which the code was written is " + rate);

      }
}
```

Input

Below is an example of Mad Libs program. This type of program allows the user to interact with the program while it is running. Find the lines of code that look new and try to determine what these lines of code accomplish.

```
import java.util.Scanner;

public class MadLib {

       public static void main(String[] args) {

              Scanner input = new Scanner(System.in);

              System.out.print("Please enter a noun: ");
              String noun = input.nextLine();

              System.out.print("Please enter an adjective: ");
              String adj = input.nextLine();

              System.out.print("Please enter an verb: ");
              String verb = input.nextLine();

              System.out.print("Please enter an integer: ");
              int num = input.nextInt();

              System.out.print("Please enter a decimal: ");
              double value = input.nextDouble();

              System.out.print("Our story is about a " + noun + ".   This particular " + noun + " likes to " + verb + ".\n");
              System.out.print("It is also very " + adj + "!  It only costs $" + value + " on Amazon.  Would you like to buy "
              + num + "?");
       }
}
```

Note: Always read in Strings before numbers! (If you want to do this in a different order, let me know and I'll tell you how to do this correctly)

Advanced: Arrays

Read through the code below. Identify the new syntax and try to interpret what it means. How is this new syntax useful? Does the new syntax have any drawbacks?

```
public class PropertyValueCalculator {

      public static void main(String[] args) {

            double[] housePrices = new double[5];

            housePrices[0] = 200000.0;
            housePrices[1] = 340000.0;
            housePrices[2] = 190000.0;
            housePrices[3] = 260000.0;
            housePrices[4] = 500000.0;

            /* The lines of code above could be simplified down to:
                  double[] housePrices = {200000.0, 340000.0, 190000.0, 260000.0, 500000.0};
            */

            double total = housePrices[0] + housePrices[1] + housePrices[2] + housePrices[3] + housePrices[4];
            double avg = total / housePrices.length;

            System.out.println("The average property value is $" + avg);
      }
}
```