Lecture 7

- Covers
  - Variables
  - Assignment
  - Expressions
  - Basic input and output
- Reading: Savitch 2.1

- Variables

Variables

- A Java variable stores an item of data
- Examples
  - a number
  - a boolean
  - a character
- The value of the data item may change as the program executes
- The value is stored in a memory location
- The name of the variable is an alias for that memory location within the program

Java program using variables

```
import java.util.*;
public class TemperatureConverter
{
   public static void main(String[ ] args)
   {
      double tempInCelsius;
      double tempInFahrenheit;
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Enter a temperature in celsius: ");
      tempInCelsius = keyboard.nextDouble( );

      tempInFahrenheit = (tempInCelsius * 9) / 5 + 32;

      System.out.print(tempInCelsius + " Celsius is equivalent to "
                                          + tempInFahrenheit);
   }
}
```

Identifiers

- Identifier = name of a class, a method, an attribute, or a variable
- Format
  - Start with a letter or the underscore symbol
  - Rest consists of letters, digits or underscore symbol
  - Examples

    sum, RATE, count, data2, Big_bonus, X_1

    height, Height, HEIGHT, _height, bigBonus
  - The dollar sign is also permitted but it is not a good idea to use it as it is usually reserved for special purposes

Identifiers

- Convention
  - Use only letters, starting with a lower case letter, with words after the first one having an initial capital

    shoeSize
  - This convention is sometimes referred to as camelBack notation
- Java is case sensitive, therefore it distinguishes between uppercase and lowercase letters in an identifier

Identifiers

- Which of the following are valid identifiers?

  X

  x34

  x*

  34xyz

  x_3_5

Keywords

- Keywords or reserved words are those that have special pre-defined meaning
- Keywords cannot be used an identifier (in particular, not as the name of a variable)
  - Examples

    int

    if

    else

Variable declarations

- A variable's type tells the compiler what sort of information the variable can contain
- The type of every variable must be declared before the variable is used
- We declare a variable by declaring its type, followed by the variable name

```
int tempInCelsius;
double totalWeight;
```

- Variables can also be initialised at declaration (assign a value to it)

```
int tempInCelsius = 0;
double totalWeight = 0.07;
```

- Primitive data types

Primitive data types

- Java has 8 primitive data types
  - byte
  - short
  - int
  - long
  - float
  - double
  - char
  - boolean

integers (byte, short, int, long)

floating-point (float, double)

numeric values (integers, floating-point)

characters (char)

truth values (boolean)

Integer data types

- Integer types store whole numbers (both positive and negative values)
- 4 types of integers in Java that store values in different amounts of memory
- The more memory they use, the larger the range of values they can store
- Use the int type (4 bytes) unless you have a specific reason for using another type

Real number data types

- Floating-point number types store numbers with fractional parts
- 2 types of floating-point numbers in Java that store values in different amounts of memory
- Not all numbers can be stored exactly using floating-point representation
- The more memory used, the larger the range of values that can be stored, and the more precise they are
- Use the double type (8 bytes) unless you have a specific reason for using another type

- Assignment

Assignment

- The assignment statement is used to set or change the value of data stored by a variable
  - Examples

```
numberOfFriends = 3;
totalWeight = 34.5;
totalWeight = singleWeight * numberOfItems;
```

Assignment

- An assignment statement has the form

  <variable> = <expression>;

- where the expression on the right is evaluated and then the resulting value is assigned to (stored in) the variable on the left
- Example

  area = width * height;

  bottles = bottles - 1;

(= is called the assignment operator)

Literals

- Each data type specifies a range of values
- For each data type, we need to know how to write its data values in a program
- The explicit data values in programs are called literals
- For example

```
singleWeight = 5.0;
```

- 5.0 is a literal of the type double

- Operations on primitive data types

Operators and operands

- An operator operates on one or more operands (arguments) and returns a value
  - Most operators are binary operators (i.e. have two operands)
  - Some are unary operators (i.e. have one operand)
  - Some operate on more than two operands

Arithmetic expressions

- Expressions may use arithmetic operators
- The basic arithmetic operators are

+	addition
-	subtraction
*	multiplication
/	division
%	remainder

Assignment

```
int a = 9;
int b = 6;
int c = 0, d = 12;
```

```
1. b = a;
2. c = 19 - b;
3. a = b - 10;
4. c = c - 2;
5. b = d * 2;
```

After	a	b	c	d
1.				
2.				
3.				
4.				
5.				

- Input & output statements

Input/output

- Input and output - used by a program for external communication
- Input
  - Read into the program
- Output
  - Written out from the program

Output

- System.out = standard output stream (monitor)
  - Examples

```
System.out.print("I am a fish");
System.out.print(numberOfFriends);
System.out.print("I toast therefore I am " +
                                numberOfFriends);
```

Output

- New lines
  - Examples

```
System.out.print("I am a fish\n");
System.out.println("I am a fish");
System.out.print("I\nam\na\nfish\n");
```

Summary

- To display information on the screen, use the object System.out and its methods print or println
- Methods print and println can take a variety of things as arguments (numbers, characters, boolean values, strings, expressions, etc.)

Input

- System.in = standard input stream (keyboard)
  - The traditional Java mechanism for reading input from the keyboard is complex: it requires you to understand about 8 different concepts to read in a single integer
  - So we initially use the Scanner class to simplify the input process
  - Examples

```
numberOfFriends = keyboard.nextInt( );

singleWeight = keyboard.nextDouble( );
```

Java program using variables

```
public class TemperatureConverter
{
    public static void main(String[ ] args)
    {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter a temperature in celsius: ");
        double tempInCelsius = keyboard.nextDouble( );

        double tempInFahrenheit = (tempInCelsius * 9) / 5 + 32;

        System.out.print(tempInCelsius
                                + " Celsius is equivalent to "
                                + tempInFahrenheit);
    }
}
```

Class exercise

- Problem
  - Write a program to read in a number and write out the number, its square, and its cube
  - Example
    - input: 4
    - output: 4 16 64

Solution

- More arithmetic operators and mathematical functions

Increment and decrement unary operators

++x	increase x by 1, return the new value of x *(increment x and then use it)*
x++	increase x by 1, return the old value of x *(use x and then increment it)*
--x	decrease x by 1, return the new value of x
x--	decrease x by 1, return the old value of x

e.g. `y = 10 + x++` *(if x is initially 2, y will be 12)*

`y = 10 + ++x` *(if x is initially 2, y will be 13)*

Pre and post increment operators

- y = 10 + x++ *is equivalent to*

```
{ y = 10 + x;
{ x = x + 1;
```

- y = 10 + ++x *is equivalent to*

```
{ x = x +1;
{ y = 10 + x;
```

Increment and decrement unary operators

- For the sake of clarity, we will only use these operators in statements by themselves
- For example

```
int count = 0;
// other statements
count++;    // equivalent to ++count and
            // count = count + 1 in this case
```

Assignment and arithmetic assignment operators

=	assignment (return the value assigned)
+=	add and assign
-=	subtract and assign
*=	multiply and assign
/ =	divide and assign
%=	calculate remainder and assign

e.g. x += y

may be used instead of x = x + y

Mathematical functions

- Most of the common functions are defined in the Math class
- For example

Math.sqrt(x)	Math.abs(x)
Math.exp(x)	Math.pow(x, y)
Math.log(x)	Math.random( )
Math.cos(x)	Math.PI

* See Savitch p.280 for more information

Class exercise

- Convert the following into Java expressions

$$\sqrt{x+y}$$

$$x^{2y}$$

$$\sin^2 x$$

$$\frac{-b+\sqrt{b^2-4ac}}{2a}$$

Java program using variables

```
public class PowerTable
{
    // Example from last lecture
    public static void main(String[ ] args)
    {
        int n = 0;
        do
        {
            n = n+1;
            System.out.println(n + " " + Math.pow(n,n));
        }
        while (n < 10);
    }
}
```

Next lecture

- Internal representation of primitive data types
- Type compatibilities and type casting
- Integer division and truncation of floating point numbers