Java Programming

Variable Syntax

- typeName variableName = value; (e.g. `int x = 2;`)
- typeName variableName; // initialize with a value later

Objects vs. Types

In Java everything is a primitive type (p. 130 in book) or an object (p.32.)

Examples of Primitive Types:

- int - whole number
- double - floating point number (decimals)
- boolean - true or false

Examples of Objects:

- Strings - String name, "Hello World!"
- Scanners - Scanner in;
- System.out

Constants

- Use constants for values that do not / cannot change and are needed in a computation
- e.g. `final double GRAVITY = 9.81;`

Operators

- Increment - `counter++;`
- Decrement - `counter--;`
- Division - / (if both numbers are ints, discards remainder) (7/4 = 1)
- Modulus - %
- Useful Math Methods - Math.sqrt(x), Math.pow(x,y), Math.ceil(x)
- Casting - change types

  ```
  double price = 3.56;
  int dollars = (int) price; // price = 3
  ```

Input/Output

- Useful to ask users for input, makes your program more flexible
- We use the Scanner class to read input from the user.

How to use the **Scanner**:

1. `import java.util.Scanner;` // include this so you can use the Scanner class
2. `Scanner input = new Scanner(System.in);` // create a Scanner object you can use to read in input from the keyboard
3. `System.out.print("Enter your age: ");` // prompt user for input
4. `int age = input.nextInt();` // define a variable to take in input, the program will wait for input and store it in the variable

Formatting output (see pg. 147) - Use printf method to format and print out multiple values

Example:

```
int x = 20;
double y = 39.97890;
System.out.printf("I am %d years old and owe $%.2f.", x, y);
```

Control Flow

Iterators:

Use an iterator to repeatedly execute a block of code until a goal is met.

While loop - executes code as long as the condition is true

```
int i = 0;

while(i < 10)
{
    i++;
    System.out.print(i);
}
```

For loops - count controlled while loop

```
for(i = 1; i < 11, i++)
{
    System.out.print(i);
}
```

These two blocks of code print the same thing.

Conditionals:

If statements

- used to implement a decision.
- If one condition is satisfied, one block of code will execute, otherwise it will be skipped and potentially a different code block will be executed.
- use "else if" if there are multiple alternatives

Example:

```
Scanner input = new Scanner(System.in);
System.out.print("How old are you? ");
int age = input.nextInt();

if(age >= 18)
{
    System.out.print("Make sure you're registered to vote!");
}

else
{
    System.out.print("When you're 18, you should register to vote.")
}
```

Programming Project Tips

Style:

- File
    - Name - same as name of your class + .java extension
    - Header - include your name, uni, file name and short description

- Formatting

    - Braces - use for if, else, for, do, while statements - be consistent!
    - Indentation - use tabs, 2 spaces, or 4 spaces (your IDE can do this for you)
    - Line Length - 80 characters or less (your IDE can do this for you)
    - White Space - seperate your code blocks

- Comments

    - really important! (especially for grading)
    - Make sure to comment above code blocks

      // short comment

      /* long

      comment */

- Naming

    - Classes - UpperCamelCase (e.g CashRegister)
    - Variables - lowerCamelCase (e.g dollars or avgPrice)
    - Methods - lowerCamelCase (e.g getPrice or toString)

- Submission - uni_project1.zip

Check out the Style Guide on Courseworks!

Debugging:

- Office Hours
- Java API
- Stack Overflow

Tip: Start early!

Examples:

1. Circle calculator - Computes circle area and circumference from radius