Notes: Introduction to Java Programming

Writing Java Code

- You write Java code as text. You then **compile** it into compiler instructions using a program called a compiler. When your java code is compiled it turns into bytecode. Then you can **run** the bytecode using the Java Runtime Environment.
- In this class you will be writing your Java code using an Integrated Development Environment (IDE) called **JGrasp**. IDEs allow you to type your code, compile your code, and run your code.

Class and Main Declarations

- Java code must be written inside of a **class**

```
public class ClassName {
}
```

- A class is executable only if it has a **main** method; the main method is the launch point of a Java program

```
public static void main(String[] args) {
}
```

Printing to the Screen

- `System.out.println("text");` prints the information inside the parentheses to the console, followed by a new line
- `System.out.print("text");` prints the information inside the parentheses to the console, without a following new line
- Note that statements/commands in Java end with semicolons

Escape sequences

`\n`	Newline
`\t`	Tab
`\"`	Double quote

`\\`	Backslash

Comments

- Comments are used to leave descriptive notes in code or to prevent code from executing:
    - Single Line comments

```
// The rest of the line after two forward slashes is ignored by the compiler
// Single line comments are… for a single line
```

- Block comments

```
/*
Everything between the opening slash-star and the closing star-slash is
ignored by the compiler. These block comments can span several lines of text.

It is important not to put a space between the slash and the star.
*/
```

- When code is put in a comment it is considered “commented out” and does not execute

```
// System.out.println(“This line doesn’t execute.”);
```

Methods

- The `main` should be short and give an good overview of what your program does (like an outline to an essay) by calling other methods
- **Declaring a method**: defines the method, but does not execute it

```
public static void sayHi() {
    System.out.println("Hi!");
}
```

- **Calling a method**: executes the method

```
sayHi();
```

- Methods can call other methods
- Methods can call themselves, but should not do so in this course
- Not all programming solutions are created equal; when writing code, you should write DRY (**D**o not **R**epeat **Y**ourself) code where possible. You do this by defining methods and then calling them multiple times.

Style

- Classes start with a capital letter, no spaces, all other words in name are also capitalized, e.g.:

```
public class TestMain
```

- Methods start with lowercase letters, no spaces, all other words in name are capitalized, e.g.:

```
public static displayRules()
```

- Indentation is important even though Java ignores it; when you open a curly brace everything inside is indented, e.g.:

```
public static void main(String[] args) {
    System.out.println("I'm indented one tab");
}
```