Introduction to Computing & Java

- Key Concept of Computing
- Main differences with Python
- Starting a program
- Output

Key Concept of Computing

- Don't need "the computer"

- Algorithm
  - Well ordered
  - Finite
  - Unambiguous
  - Doable

  } set of instructions

Key Concept of Computing - Algorithm

- Instructions
  - **Well ordered**
  - **Finite**
  - Unambiguous
  - Doable

Key Concept of Computing - Algorithm

- Instructions
  - Well ordered
  - Finite
  - **Unambiguous**
  - **Doable**

Algorithm

- To sort a list of numbers from biggest to smallest

- E.g.

6	-3	21	18	10	45	1	-17	56	34

Algorithm – Sort Numbers Big to Small

E.g.

6	-3	21	18	10	45	1	-17	56	34

What Do We Need in a Programming Language?

- A way to store information
- Instructions, meeting definition of algorithm, including:

Main Differences with Python

- Harder to make a Java program “run”
- Java is “lower” in level than Python

Java

- Developed at Sun Computers (Oracle) in 1990s

- Originally called Oak, later given the name Java

Java

- Now mainly used for running stuff on Internet or Android phone
- Originally meant to be a programming language for embedded systems

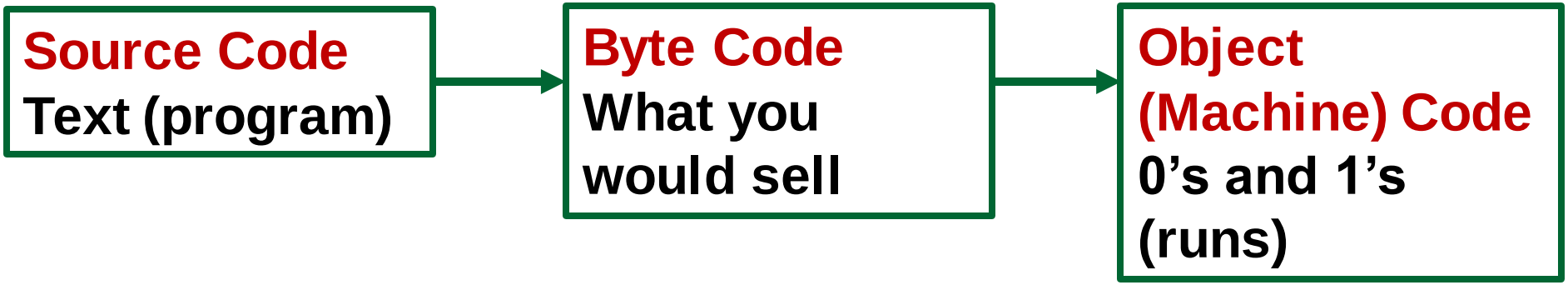
- Claims to Fame:

Java is Compiled*ish*

- Vs. Python which is **interpreted**

- A line at a time
- Or the whole file

Java is Compiled



First Java Program

```
public class FirstProgram {
    /**
     * method simply to illustrate starting Java
     * concepts.
     * @param args - the command line arguments (none)
     */
    public static void main(String[ ] args) {
        int x;
        x = 5;  //x is the favorite number
        System.out.println("Hi There");
        System.out.println("Favorite number is: " + x);
    } //main
} //class FirstProgram
```

First Java Program continued

```
public class FirstProgram {
    /**
     * method simply to illustrate starting Java
     * concepts.
     * @param args - the command line arguments (none)
     */
    public static void main(String[ ] args) {
        int x;
        x = 5;  //x is the favorite number
        System.out.println("Hi There");
        System.out.println("Favorite number is: " + x);
    } //main
} //class FirstProgram
```

First Java Program continued

```
public class FirstProgram {
    /**
     * method simply to illustrate starting Java
     * concepts.
     * @param args - the command line arguments (none)
     */
    public static void main(String[ ] args) {
        int x;
        x = 5;  //x is the favorite number
        System.out.println("Hi There");
        System.out.println("Favorite number is: " + x);
    } //main
} //class FirstProgram
```

First Java Program continued

```
public class FirstProgram {
    /**
     * method simply to illustrate starting Java
     * concepts.
     * @param args - the command line arguments (none)
     */
    public static void main(String[ ] args) {
        int x;
        x = 5;  //x is the favorite number
        System.out.println("Hi There");
        System.out.println("Favorite number is: " + x);
    } //main
} //class FirstProgram
```

First Java Program continued

```
public class FirstProgram {
    /**
     * method simply to illustrate starting Java
     * concepts.
     * @param args - the command line arguments (none)
     */
    public static void main(String[ ] args) {
        int x;
        x = 5;  //x is the favorite number
        System.out.println("Hi There");
        System.out.println("Favorite number is: " + x);
    } //main
} //class FirstProgram
```

First Java Program continued

```
public class FirstProgram {
    /**
     * method simply to illustrate starting Java
     * concepts.
     * @param args - the command line arguments (none)
     */
    public static void main(String[ ] args) {
        int x;
        x = 5;  //x is the favorite number
        System.out.println("Hi There");
        System.out.println("Favorite number is: " + x);
    } //main
} //class FirstProgram
```

IntelliJ

- IDE for Java

Making a Java Project in IntelliJ

- Start IntelliJ
- New Project
  - Enter name and location
  - Build system: IntelliJ
  - NOTE: do NOT enable “AI Assistant”

- You should now have the start of a Java file in which to type your program. The file will be called Main.java

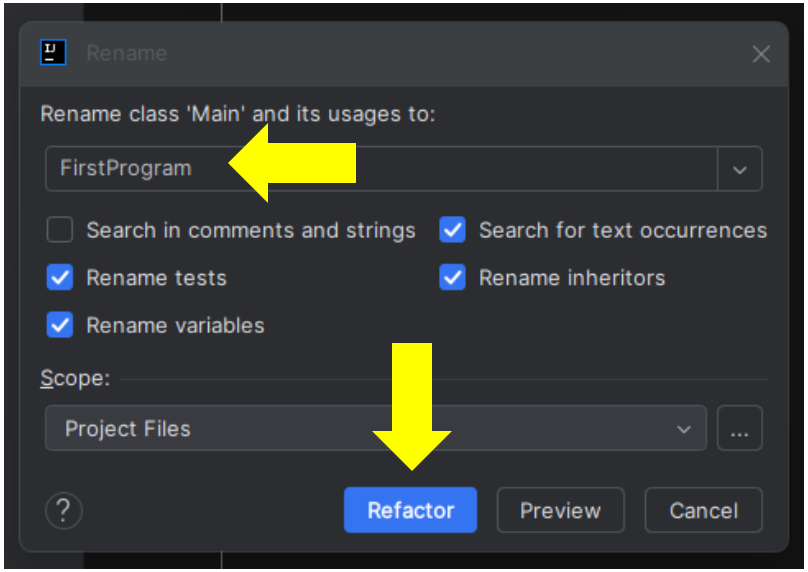
IntelliJ Personalization

- Change settings, e.g. font size
  - Ctrl-Alt “os”
- Keep top menu bar visible
  - View → Appearance → Show Main Menu in Separate Tool Bar
- Change file name
  - Click to highlight the class name, then from the top menu tabs select Refactor

IntelliJ Personalization continued

- Change file name
  - Click to highlight the class name
  - Then right-click and select “Rename”
  - Then right-click again and select “Rename” again
  - See this dialog:

2X



  - Type the new file name and click on Refactor

First Java Program: Put into File

Put the program into an actual file (called FirstProgram.java) in IntelliJ

Syntax Errors

- Note that IntelliJ gives syntax help as you type
- Still learn the syntax yourself!
- What is a syntax error?

Running a Java File in IntelliJ

- Shift-F10
- or Run menu; then Run <filename>

- A new window will appear with your results

Write a Program – Simple Sum

- Write a Java program to sum the integers from 1 up to n, where n is a parameter.
- If n is negative or 0, just answer 0.

Simple Sum Program

- Put your program here:

Write a Program – Multiples

- Write a Java method to take an integer and a bound, then print all multiples of the integer up to, but not exceeding, the bound.
- If the multiple is negative, then print all multiples down to the bound.
- Watch error conditions, for example
  - If the multiple is 0
  - If the multiple is negative, but the bound is positive
  - Etc.

Example, with 3 up to 40:

```
3 6 9 12 15 18 21 24 27 30 33 36 39
```

Example, with -3 down to -20:

```
-3 -6 -9 -12 -15 -18
```

Multiples Method

- Put your method here:

The End

Review

- Which programming language is a higher-level language—Java or Python? Why?
- What is machine code?
- What is byte code?
- What are Java's 2 main strengths as a programming language?

Review

- What line of code must always be present somewhere in a Java project?
- What does `public` mean?
- If your Java program is in the file "`SuperSunny.java`", what line of code containing the keyword "`class`" would be in your file?