

Computer Programming Fundamentals

CS 152

Professor: Leah Buechley

TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza

Time: MWF 10:00-10:50am

https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/

WEBSITE

https://handandmachine.cs.unm.edu/classes/CS152_Fall2021

Syllabus

Schedule

with class slides, assignments, etc.

Policies

Links to other material

USE PIAZZA FOR QUESTIONS DURING LECTURE

- We'll use the live chat feature
- Post questions or issues you're having
- Up vote and down vote other posts
- I will check in periodically during lecture
- Also feel free to raise your hand
- Please don't interrupt and wait until you are called on to ask a question.

DUE MONDAY: ASSIGNMENT 1

- Due Monday 8/30 by 9:30am
- Essay: What Excites you about Computing?
- Include an example of a person or project that you find inspiring.
- Submit via UNM Learn

LETS START PROGRAMMING

IntelliJ

**IF YOU'RE ON A CHROMEBOOK or IPAD
OPEN UP REPLIT**

SETTING UP YOUR FIRST PROJECT

**OPEN UP FirstProgram PROJECT
OR
START FROM SCRATCH**

WE'LL USE A DIFFERENT PROCESS

DON'T JUMP AHEAD

The next steps are important to get just right.
They're easy to mess up.
Be patient :)

DON'T JUMP AHEAD


IntelliJ does a lot behind the scene

Creates a lot of files and folders

The project structure can get messed up

IntelliJ PROJECT SETUP

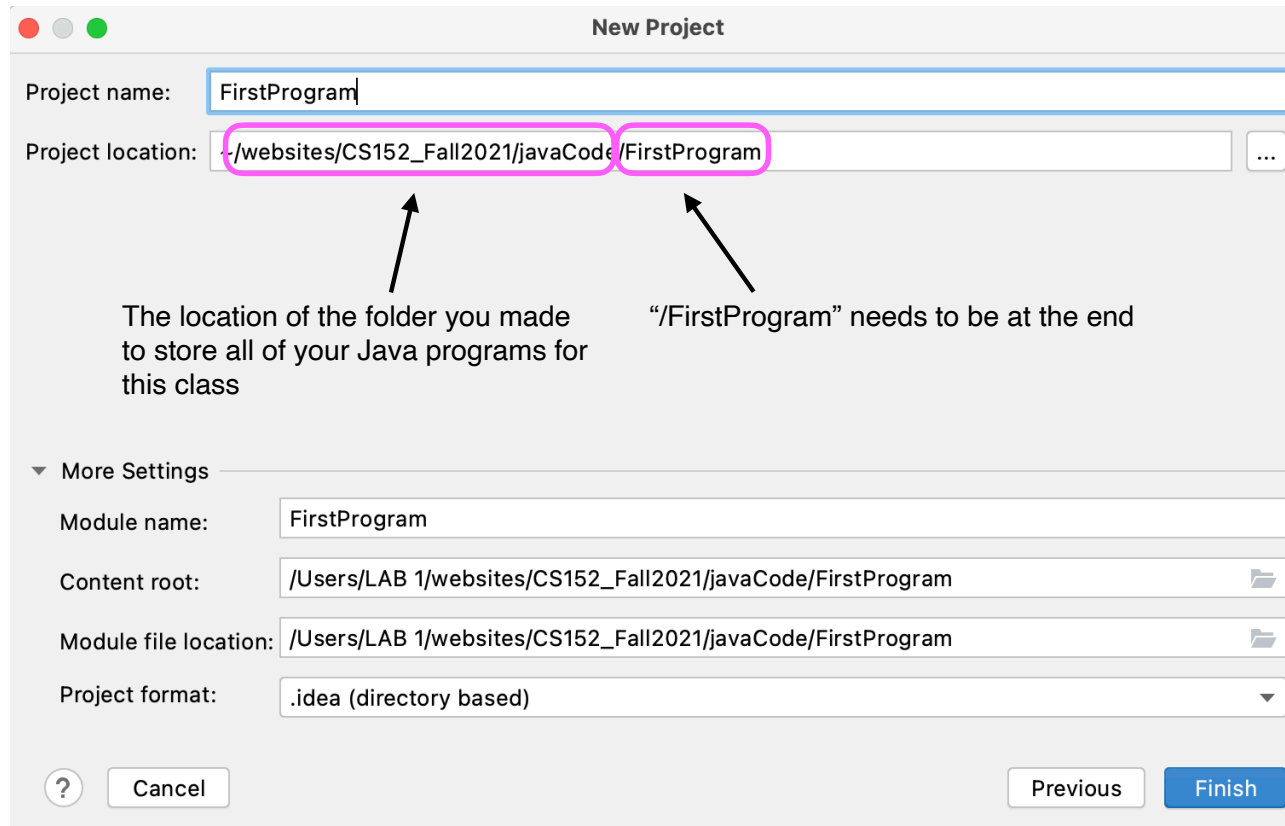
1. Create a new project
2. Select your newly downloaded SDK
3. Click “Next”
4. Click “Next”

Project SDK:  openjdk-16 java version "16.0.2"

IntelliJ PROJECT SETUP

1. Under Project location, browse to a location you will remember. This is where you'll store all your Java programs. Create a new folder called "CS152Java" or something similar
2. For PCs:
`C:\Users\YOUR_ACCOUNT\CS152Java\FirstProgram`
3. Type in "FirstProgram" for Project name
4. IMPORTANT: Make sure "FirstProgram" is at the end of the text you see in Project location. Add this text if it isn't there. This creates a folder for your new project called FirstProgram.

IntelliJ PROJECT SETUP



The screenshot shows the 'New Project' dialog in IntelliJ IDEA. The 'Project name' field contains 'FirstProgram'. The 'Project location' field contains the path './websites/CS152_Fall2021/javaCode/FirstProgram', with the directory part and the final folder name circled in pink. Two arrows point from explanatory text below to these circled parts. The 'More Settings' section is expanded, showing 'Module name' as 'FirstProgram', 'Content root' and 'Module file location' as '/Users/LAB 1/websites/CS152_Fall2021/javaCode/FirstProgram', and 'Project format' as '.idea (directory based)'. Buttons for 'Cancel', 'Previous', and 'Finish' are visible at the bottom.

New Project

Project name: FirstProgram

Project location: ./websites/CS152_Fall2021/javaCode/FirstProgram

The location of the folder you made to store all of your Java programs for this class

“/FirstProgram” needs to be at the end

More Settings

Module name: FirstProgram

Content root: /Users/LAB 1/websites/CS152_Fall2021/javaCode/FirstProgram

Module file location: /Users/LAB 1/websites/CS152_Fall2021/javaCode/FirstProgram

Project format: .idea (directory based)

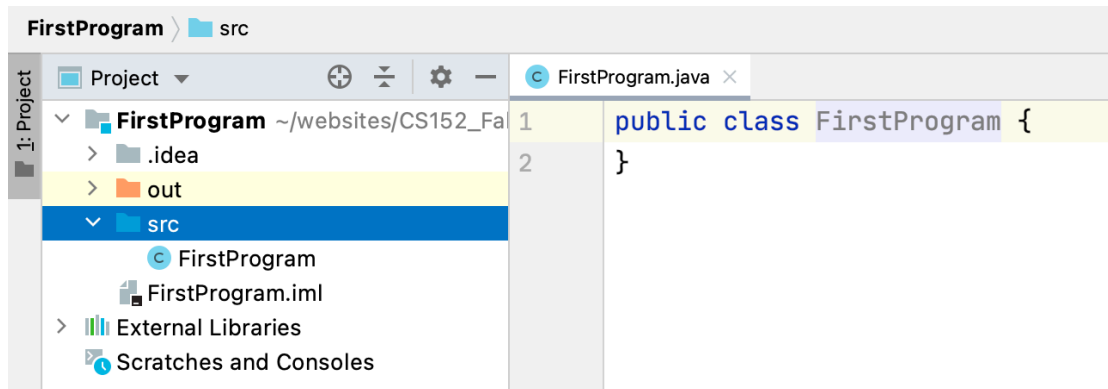
Cancel Previous Finish

IntelliJ PROJECT SETUP

1. Find the CS152Java folder on your computer
2. For PCs:
C:\Users\YOUR_ACCOUNT\CS152Java\FirstProgram

CREATE A NEW FILE

- Create a new Java Class file in the src directory.
directory = folder, src = “source code”
- Name it “FirstProgram.java”
- This will generate the basic code structure
(see below)



The screenshot shows an IDE window titled "FirstProgram" with a sub-window for the "src" directory. The file explorer on the left shows the project structure, including the "src" folder which contains a new file named "FirstProgram.java". The code editor on the right shows the following code:

```
1 public class FirstProgram {  
2 }
```

JAVA PROGRAMS

- Class name “FirstProgram” must match file name “FirstProgram.java”
- Class name should start with a capital letter
- If name is more than 2 words, all words are capitalized. No spaces. “FirstProgram”, “GreenApple”, etc.

ADD A MAIN METHOD

- Type “main” inside of the curly brackets
- Hit return and notice how the structure for a piece of code automatically pops up
- This is the main method

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

MAIN METHOD

- Every Java program has a main() method
- Always has the same structure and syntax:

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

- This is the piece of code that runs when your program runs.

ADD A PRINT STATEMENT

- Add a line that will print “Hello World!”
- `System.out.println(“Hello World!”)`

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

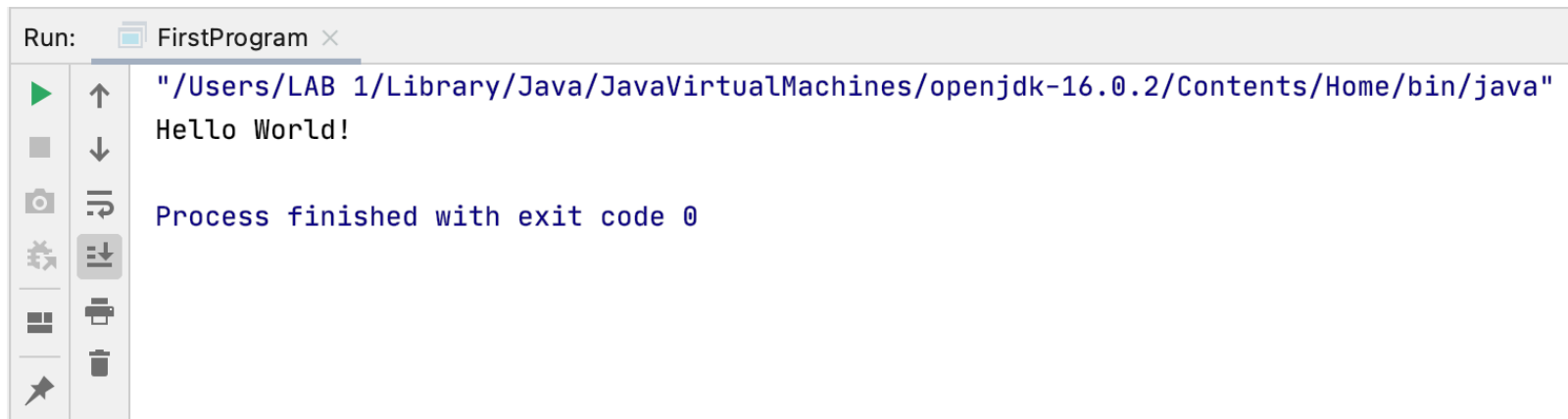
COMPILE & RUN YOUR PROGRAM

- Click on the green arrow next to the main method
- Click “Run FirstProgram.main()”

```
1 ▶ public class FirstProgram {  
2  
3 ▶ public static void main(String[] args) {  
4     System.out.println("Hello World!");  
5 }  
6 }
```

COMPILE & RUN YOUR PROGRAM

- Look at the bottom half of your screen



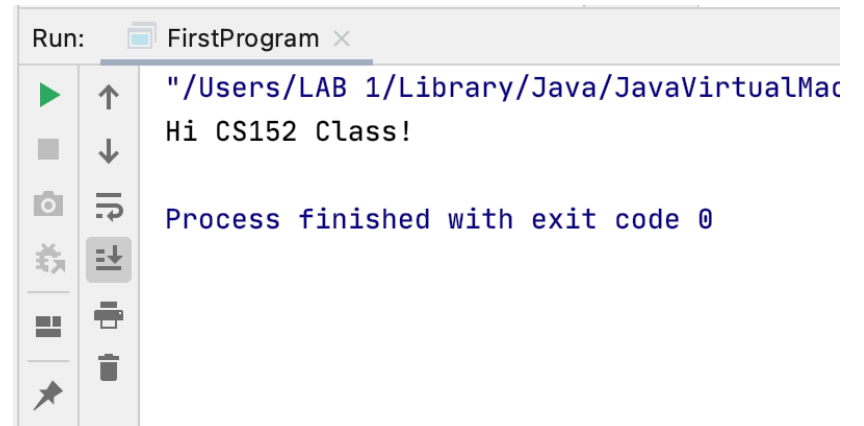
The screenshot shows a 'Run' console window with a tab labeled 'FirstProgram'. The console output is as follows:

```
"/Users/LAB 1/Library/Java/JavaVirtualMachines/openjdk-16.0.2/Contents/Home/bin/java"  
Hello World!  
  
Process finished with exit code 0
```

The console window includes a vertical toolbar on the left with icons for play, stop, refresh, and other actions.

PLAY WITH THE TEXT THAT'S PRINTED

```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```



The screenshot shows a window titled "Run: FirstProgram x". The output text is as follows:

```
"/Users/LAB 1/Library/Java/JavaVirtualMac  
Hi CS152 Class!  
  
Process finished with exit code 0
```

The console window includes a toolbar on the left with icons for running, stepping through, and other debugging actions.

WRITING CODE

- Important strategy
- Try to ignore stuff that doesn't make sense
- It just has to be there for the program to work
- The fact that it doesn't make sense is because Java isn't an ideal programming language
- A better language would make more sense!
- Focus on the part you can control and understand

WRITING CODE: FOCUS

```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

This says write something on a line
print = write
ln = line

The stuff in green is what gets written

WRITING CODE: FOCUS, STRUCTURE

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

Notice two curly brackets at the beginning and end of the program
These tell the compiler when FirstProgram begins and ends

WRITING CODE: FOCUS, STRUCTURE

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

Notice two curly brackets at the beginning and end of the main() method
These tell the compiler when the main() method begins and ends

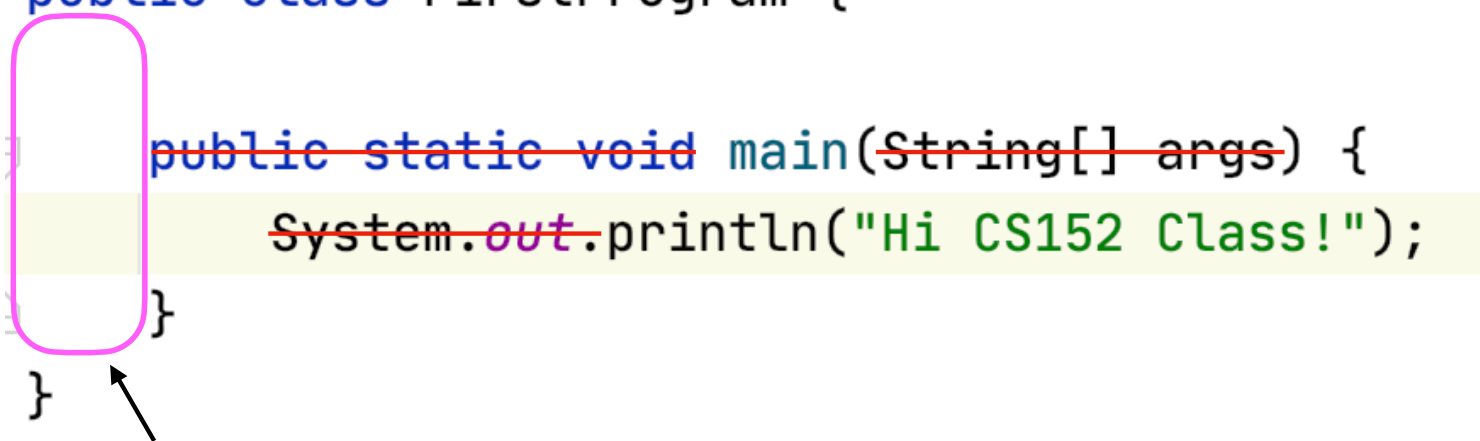
WRITING CODE: FOCUS, STRUCTURE

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

↑
semicolon at the end of each statement
statements are inside curly brackets
a period for programming in Java

WRITING CODE: FOCUS, STRUCTURE

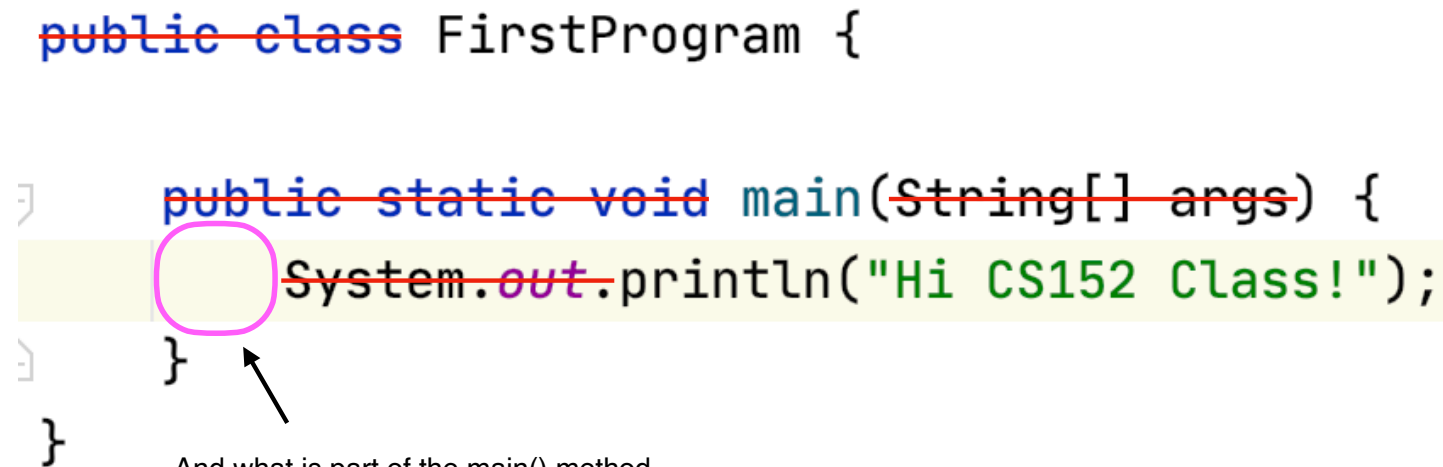
```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

A diagram illustrating code structure. The code is shown with several annotations: a pink rounded rectangle highlights the opening curly brace of the class, the opening curly brace of the main method, and the closing curly brace of the main method. A black arrow points from the closing curly brace of the main method to the closing curly brace of the class. The text 'System.out.println' is highlighted in green, and the string 'Hi CS152 Class!' is also highlighted in green. The text 'public class', 'public static void', and 'String[]' are crossed out with red lines.

Indentations help you keep track of what is part of FirstProgram

WRITING CODE: FOCUS, STRUCTURE

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```



And what is part of the main() method

questions?

PLAY WITH THE TEXT THAT'S PRINTED

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

PLAY WITH THE TEXT THAT'S PRINTED

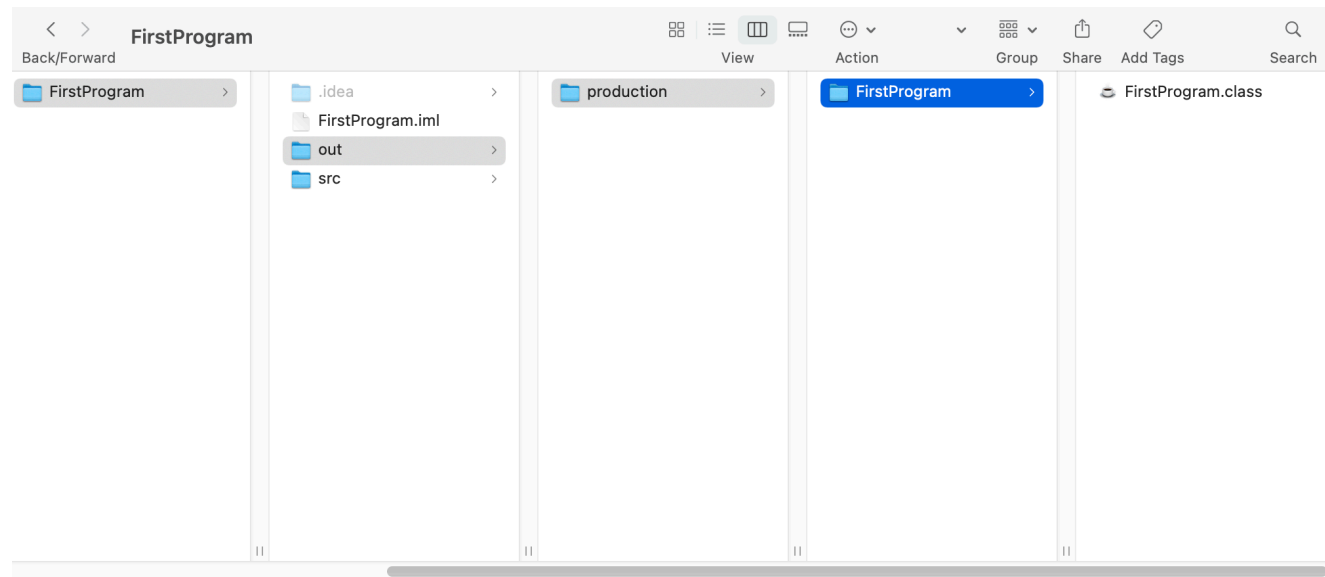
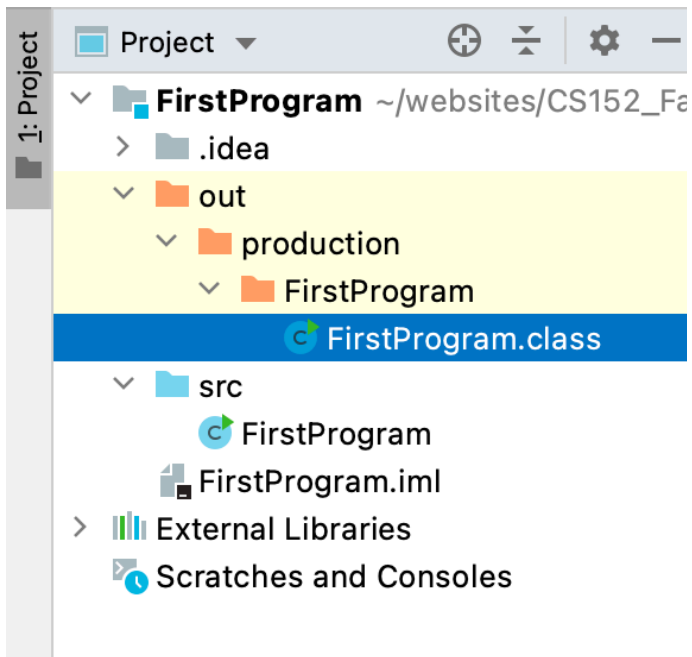
```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.println("Hi CS152 Class!");  
        System.out.println("How's it going?");  
    }  
}
```

```
}
```

COMPILING & RUNNING JAVA PROGRAMS


- Source code written in a .java file
- Compiler converts source code into byte code, a .class file
- The Java Virtual Machine (JVM) runs this byte code, translating it into instructions that your computer executes

FIND THE .CLASS FILE



COMMENTS

COMMENTS

```
public class FirstProgram {  
    /*  
     this is a comment that  
        is more than one line  
    */  
    public static void main(String[] args) {  
        //this is a comment on a single line  
        System.out.println("Hi CS152 Class!");  
    }  
}
```

COMMENTS

- A way to write yourself notes in the code
- Ignored by the compiler
- Multiline comments: `/* comment text */`
- Single line comments: `// comment text`

ADD SOME COMMENTS TO YOUR CODE

```
public class FirstProgram {  
  
    //this is the main method  
    public static void main(String[] args) {  
        //this is my nifty print statement  
        System.out.println("Hi CS152 Class!");  
    }  
  
}
```


questions?

A LITTLE COMPUTING

- Make a Fahrenheit to Celsius converter
- Let's look up today's high temperature
- And print it on the screen

PRINT OUT A NUMBER

```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println(93);  
    }  
}
```

PRINT VS PRINTLN

```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println("Today's high temperature will be: ");  
        System.out.println(93);  
    }  
}
```

Today's high temperature will be:
93

```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.print("Today's high temperature will be: ");  
        System.out.println(93);  
    }  
}
```

Today's high temperature will be: 93

PRINT VS PRINTLN

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        System.out.print("Today's high temperature will be ");  
        System.out.print(93);  
        System.out.println(" degrees F.");  
    }  
  
}
```

Today's high temperature will be 93 degrees F.

F to C CONVERTER

- Celsius = $(93F - 32) * 5/9$

A CONVERTER PROGRAM

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        //print temperature in Fahrenheit  
        System.out.print("Today's high temperature will be ");  
        System.out.print(93);  
        System.out.println(" degrees F.");  
  
        //print temperature in Celsius  
        System.out.print("That's ");  
        System.out.print((93-32)*5/9);  
        System.out.println(" in degrees C.");  
    }  
}
```

Today's high temperature will be 93 degrees F.
That's 33 in degrees C.

NOTE COMMENTS!

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        //print temperature in Farenheit  
        System.out.print("Today's high temperature will be ");  
        System.out.print(93);  
        System.out.println(" degrees F.");  
  
        //print temperature in Celsius  
        System.out.print("That's ");  
        System.out.print((93-32)*5/9);  
        System.out.println(" in degrees C.");  
    }  
}
```

Today's high temperature will be 93 degrees F.
That's 33 in degrees C.

IF WE WANT TO CHANGE THE TEMP?

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        //print temperature in Fahrenheit  
        System.out.print("Today's high temperature will be ");  
        System.out.print(93);  
        System.out.println(" degrees F.");  
  
        //print temperature in Celsius  
        System.out.print("That's ");  
        System.out.print((93-32)*5/9);  
        System.out.println(" in degrees C.");  
    }  
}
```

Today's high temperature will be 93 degrees F.
That's 33 in degrees C.

IF WE WANT TO CHANGE THE TEMP?

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        //print temperature in Fahrenheit  
        System.out.print("Today's high temperature will be ");  
        System.out.print(93);  
        System.out.println(" degrees F.");  
  
        //print temperature in Celsius  
        System.out.print("That's ");  
        System.out.print((93-32)*5/9);  
        System.out.println(" in degrees C.");  
    }  
}
```

Today's high temperature will be 93 degrees F.
That's 33 in degrees C.

Have to change the value in two places.

ADD A VARIABLE

```
public static void main(String[] args) {  
    int temperatureF = 93;  
}
```

VARIABLES IN JAVA

variable's type

```
int temperatureF = 93;
```

int = integer

a whole number

variable's name

```
int temperatureF = 93;
```

variable's value

```
int temperatureF = 93;
```


semicolon

```
int temperatureF = 93;
```

USE THE VARIABLE

```
public static void main(String[] args) {  
    int temperatureF = 93;  
  
    //print temperature in Fahrenheit  
    System.out.print("Today's high temperature will be ");  
    System.out.print(temperatureF);  
    System.out.println(" degrees F.");  
  
    //print temperature in Celsius  
    System.out.print("That's ");  
    System.out.print((temperatureF-32)*5/9);  
    System.out.println(" in degrees C.");  
}
```

Today's high temperature will be 93 degrees F.
That's 33 in degrees C.

Now we only need to change the value in one place.

CHANGE THE VALUE OF THE VARIABLE

```
public static void main(String[] args) {  
    int temperatureF = 85;  
  
    //print temperature in Fahrenheit  
    System.out.print("Today's high temperature will be ");  
    System.out.print(temperatureF);  
    System.out.println(" degrees F.");  
  
    //print temperature in Celsius  
    System.out.print("That's ");  
    System.out.print((temperatureF-32)*5/9);  
    System.out.println(" in degrees C.");  
}
```

Today's high temperature will be 85 degrees F.
That's 29 in degrees C.

ADD ANOTHER VARIABLE

```
public static void main(String[] args) {  
    int temperatureF = 85;  
    int temperatureC = (temperatureF-32)*5/9;  
}
```

USE THE VARIABLE

```
public static void main(String[] args) {  
    int temperatureF = 85;  
    int temperatureC = (temperatureF-32)*5/9;  
  
    //print temperature in Fahrenheit  
    System.out.print("Today's high temperature will be ");  
    System.out.print(temperatureF);  
    System.out.println(" degrees F.");  
  
    //print temperature in Celsius  
    System.out.print("That's ");  
    System.out.print(temperatureC);  
    System.out.println(" in degrees C.");  
}
```

Today's high temperature will be 85 degrees F.
That's 29 in degrees C.

A NICE CLEAR PROGRAM

```
public static void main(String[] args) {  
    int temperatureF = 85;  
    int temperatureC = (temperatureF-32)*5/9;  
  
    //print temperature in Fahrenheit  
    System.out.print("Today's high temperature will be ");  
    System.out.print(temperatureF);  
    System.out.println(" degrees F.");  
  
    //print temperature in Celsius  
    System.out.print("That's ");  
    System.out.print(temperatureC);  
    System.out.println(" in degrees C.");  
}
```

questions?

COMPILE ERRORS

COMPILE ERRORS

```
public static void main(String[] args) {  
    int temperatureF = 85;  
    int temperatureC = (temperatureF-32)*5/9;  
  
    //print temperature in Fahrenheit  
    System.out.print("Today's high temperature will be ");  
    System.out.print(temperatureF);  
    System.out.println(" degrees F.");  
  
    //print temperature in Celsius  
    System.out.print("That's ");  
    System.out.print(temperatureC);  
    System.out.println(" in degrees C.");  
}
```

COMPILE ERRORS

```
public static void main(String[] args) {
    int temperatureF = 85;
    int temperatureC = (temperatureF-32)*5/9;

    //print temperature in Fahrenheit
    System.out.print("Today's high temperature will be ")0
    System.out.print(temperatureF);
    System.out.println(" degrees F.");

    //print temperature in Celsius
    System.out.print("That's ");
    System.out.print(temperatureC);
    System.out.println(" in degrees C.");
}
```

[/Users/LAB 1/websites/CS152_Fall2021/javaCode/FirstProgram/src/FirstProgram.java:8:62](#)

java: ';' expected

COMPILE ERRORS

- Errors in the syntax of your program
- Like spelling and punctuation mistakes when you write
- The compiler is picky and not super smart!
- The program won't compile if there are any errors in your code

COMPILE ERRORS

```
public static void main(String[] args) {  
    int temperatureF = 85;  
    int temperatureC = (temperatureF-32)*5/9;  
  
    //print temperature in Fahrenheit  
    System.out.print("Today's high temperature will be ");  
    System.out.print(temperatureF);  
    System.out.println(" degrees F.");  
  
    //print temperature in Celsius  
    System.out.print("That's ");  
    System.out.print(temperatureC);  
    System.out.println(" in degrees C.");  
}
```

COMPILE ERRORS

```
public static void main(String[] args) {
    int temperatureF = 85;
    int temperatureC = (temperatureF-32)*5/9;

    //print temperature in Fahrenheit
    System.out.print("Today's high temperature will be ");
    System.out.print(temperatureF);
    System.out.println(" degrees F.");

    //print temperature in Celsius
    System.out.print("That's ");
    System.out.print(temperatureC);
    System.out.println(" in degrees C.");
}
```

[/Users/LAB 1/websites/CS152_Fall2021/javaCode/FirstProgram/src/FirstProgram.java:9:26](#)

```
java: cannot find symbol
  symbol:   variable temperatureF
  location: class FirstProgram
```

COMPILE ERRORS

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        int temperatureF = 85;  
        int temperatureC = (temperatureF-32)*5/9;  
  
        //print temperature in Fahrenheit  
        System.out.print("Today's high temperature will be ");  
        System.out.print(temperatureF);  
        System.out.println(" degrees F.");  
  
        //print temperature in Celsius  
        System.out.print("That's ");  
        System.out.print(temperatureC);  
        System.out.println(" in degrees C.");  
    }  
}
```

COMPILE ERRORS

```
public class FirstProgram {  
  
    public static void main(String[] args) {  
        int temperatureF = 85;  
        int temperatureC = (temperatureF-32)*5/9;  
  
        //print temperature in Fahrenheit  
        System.out.print("Today's high temperature will be ");  
        System.out.print(temperatureF);  
        System.out.println(" degrees F.");  
  
        //print temperature in Celsius  
        System.out.print("That's ");  
        System.out.print(temperatureC);  
        System.out.println(" in degrees C.");  
    }  
}
```

0

/Users/LAB_1/websites/CS152_Fall2021/javaCode/FirstProgram/src/FirstProgram.java:16:6
java: reached end of file while parsing

COMPILE ERRORS

- You'll always have compile errors
- Part of the programming process
- You'll get better & better at finding and fixing them
- IntelliJ will help you (most of the time):
 - telling you where the error is
 - changing the color of text where there is a problem
 - moving your cursor to the location of the error
 - printing error messages
 - etc.

questions?

DUE MONDAY: ASSIGNMENT 1

- Due Monday 8/30 by 9:30am
- Essay: What Excites you about Computing?
- Include an example of a person or project that you find inspiring.
- Submit via UNM Learn

Thank you!

CS 152

Professor: Leah Buechley

TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza

Time: MWF 10:00-10:50am

https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/