Faculty of Arts & Sciences
Department of Computer Science
CMPS 252—Software Construction
Spring 2019–2020

HW0: Due Wed 29/01/2020 @ 9:00 AM

Part I: Typing, compiling, and running your first program.

1- ☐ Create a folder on the filesystem to save your work. Make it easy and use a short path `Z:\FJP` (stands for First Java Program)

2- ☐ Open notepad (do not use eclipse or any other IDE or code editor) and type the following code segment

```
Public class SomeClass {
        public static void Main(String[] args) {
                System.out.println("Hello World");
        }
}
```

3- ☐ Save the class as `z:\FJP\FirstClass.java` and keep notepad open

4- ☐ Go to a command prompt and find where the JDK is installed (usually in `Z:\Program Files\Java\jdk1.x.x`). If Java 13 is installed then you'll find `jdk13.0.2`...

5- Type cd `z:\FJP ⏎` (this symbol is an indicator for you to press enter)

6- ☐ Type `javac ⏎` (this is the java compiler)

7- ☐ If you do not get an error proceed to Step 11

8- You'll get an error *'javac' is not recognized as an internal or external command, operable program or batch file.* This means the command interpreter does not recognize `javac.exe` (or `.cmd` or `.bat` …) as a valid command. Even though the JDK is installed, the directory/folder where `javac.exe` resides is not part of the path searched by the command line interpreter (CLI). This directory is `Z:\Program Files\Java\jdk13.0.2\bin` and it houses all the tools and utilities that the JDK provides including the compiler, debugger, interpreter, etc.

9- ☐ You need to add the path where javac.exe is located to the search path of your CLI. To do so, type `path=%path%;"z:\Program Files\Java\jdk13.0.2\bin" ⏎`

10- ☐ Now try running the Java compiler again. Confirm that it works (you'll get a usage statement)

11- ☐ Now ask it to compile the program you just typed by typing `javac FirstClass.java⏎`

12- ☐ You'll get an error (notice the caret under the P). Change `Public` to `public`, save the file and try step 11 again.

13- ☐ You'll get an error. Read the message. Do you know what you need to do? Change `SomeClass` to `FirstClass`, save the file and try step 11 again.

14- ☐ This time it compiles successfully (no error message is displayed).

15- ☐ Try running the program by typing `java FirstClass.class⏎`

16- ☐ You'll get an error since it is not able to find the class named `FirstClass.class`

17- ☐ Try running the program by typing `java FirstClass ⏎`

18- ☐ This time it runs but it will fail telling you that `Main` method is not available.

19- ☐ Change `Main` to `main` in your source code, save the file and try step 11 again.

20- ☐ Try running the program again (step 17). This time it runs successfully.

Part II: Packaging multiple .class files into a JAR file

1- ☐ In the same folder (`z:\FJP`) create a second source code file:

```
public class SecondClass {
	public static void main(String[] args) {
		for(int i=0;i<10;i++)
		{
			char c = (char)(65+i);
			System.out.println("Greeting World " + c);
		}
	}
}
```

2- ☐ Save and compile this file.

3- ☐ Using the `jar.exe` tool create a jar file that houses both the .class files you created:

```
jar -cfe MyProgram.jar FirstClass FirstClass.class SecondClass.class⏎
```

`c`: tells `jar.exe` to create a new jar file

`f`: tells `jar.exe` to output to a specific jar file (`MyProgram.jar`)

`e`: tells `jar.exe` to mark the class `FirstClass` as the class to execute (entrypoint)

Confirm that you have created the jar file: type `dir *.jar⏎` and confirm the existence of the `MyProgram.jar`

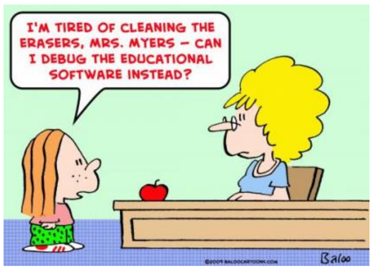
4- ☐ Run the program: `java -jar MyProgram.jar⏎` you should get the string `Hello World`

5- ☐ Recreate the jar file this time making `SecondClass` the entrypoint of the jar file

6- ☐ Run the program and confirm that the output is

```
Greeting World A
Greeting World B
Greeting World C
Greeting World D
Greeting World E
Greeting World F
Greeting World G
Greeting World H
Greeting World I
Greeting World J
```

Part III: Debugging using CLI


I'M TIRED OF CLEANING THE ERASERS, MRS. MYERS – CAN I DEBUG THE EDUCATIONAL SOFTWARE INSTEAD?
Baloo

1- ☐ To debug `SecondClass` using the `jdb.exe` tool, you need to compile the class using the `-g option`: type `javac -g SecondClass.java⏎`

2- ☐ Now start debugging, type `jdb SecondClass⏎`

3- ☐ The debugger is ready to load and run your code

4- ☐ Type `run⏎`. The debugger will run your program and you should see the same result as previous. However this is not interesting!

5- ☐ Again, type `jdb SecondClass⏎`

6- ☐ Tell the debugger to set a breakpoint (stop) at line 5: type `stop at SecondClass:5⏎`

7- ☐ Now type `run⏎` again, you'll see that the debugger executed the lines up to and not including line 5

8- ☐ To inspect the local variable: type `print i⏎` you'll see that `i` is zero at this stage.

9- ☐ Type `step⏎` to go to the next line. Now you ran line 5 and are stopped at line 6.

10- ☐ Type `print c⏎`. You'll see that C is A (ASCII for 65)

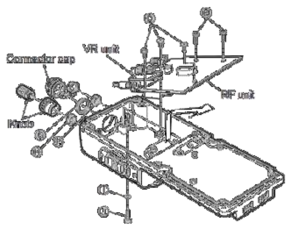
11- ☐ Type `locals⏎` to see all the local variables.

12- ☐ Type `list⏎` to see the source code and the line you are currently on.

13- ☐ Now clear the breakpoints: type `clear SecondClass:5⏎` then `cont⏎` and the program continues running all the way through.

Part IV: Disassemble the Java bytecode

1- ☐ The java disassembler is `javap.exe`. Type `javap -c FirstClass.class`. Locate the corresponsding Java bytecode.



```
    Z:\FJP>javap -v  FirstClass.class
Classfile /C:/FJP/FirstClass.class
  Last modified Feb 20, 2012; size 425 bytes
  MD5 checksum 34118d5a2e9a83a6350e22b92047c2d1
  Compiled from "FirstClass.java"
public class FirstClass
  SourceFile: "FirstClass.java"
  minor version: 0
  major version: 51
  flags: ACC_PUBLIC, ACC_SUPER

Constant pool:
   #1 = Methodref          #6.#15         //  java/lang/Object."<init>":()V
   #2 = Fieldref           #16.#17        //  java/lang/System.out:Ljava/io/PrintStream;
   #3 = String             #18            //  Hello World
   #4 = Methodref          #19.#20        //  java/io/PrintStream.println:(Ljava/lang/String;)V
   #5 = Class              #21            //  FirstClass
   #6 = Class              #22            //  java/lang/Object
   #7 = Utf8               <init>
   #8 = Utf8               ()V
   #9 = Utf8               Code
  #10 = Utf8               LineNumberTable
  #11 = Utf8               main
  #12 = Utf8               ([Ljava/lang/String;)V
  #13 = Utf8               SourceFile
  #14 = Utf8               FirstClass.java
  #15 = NameAndType        #7:#8          //  "<init>":()V
  #16 = Class              #23            //  java/lang/System
  #17 = NameAndType        #24:#25        //  out:Ljava/io/PrintStream;
  #18 = Utf8               Hello World
  #19 = Class              #26            //  java/io/PrintStream
  #20 = NameAndType        #27:#28        //  println:(Ljava/lang/String;)V
  #21 = Utf8               FirstClass
  #22 = Utf8               java/lang/Object
  #23 = Utf8               java/lang/System
  #24 = Utf8               out
  #25 = Utf8               Ljava/io/PrintStream;
  #26 = Utf8               java/io/PrintStream
  #27 = Utf8               println
  #28 = Utf8               (Ljava/lang/String;)V
{
  public FirstClass();
    flags: ACC_PUBLIC

    Code:
      stack=1, locals=1, args_size=1
         0: aload_0
         1: invokespecial #1                  // Method java/lang/Object."<init>":()V
         4: return
      LineNumberTable:
        line 1: 0

  public static void main(java.lang.String[]);
    flags: ACC_PUBLIC, ACC_STATIC

    Code:
      stack=2, locals=1, args_size=1
         0: getstatic     #2                  // Field
java/lang/System.out:Ljava/io/PrintStream;
         3: ldc           #3                  // String Hello World
         5: invokevirtual #4                  // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
         8: return
      LineNumberTable:
        line 3: 0
        line 4: 8
}
```

Java bytecode

Part V: Generating online documentation

1- ☐ Add the following comments to FirstClass.java then save the file

```
/** This is the first class I wrote in CMPS252 Spring 2020
 *
 * @author Mahmoud Bdeir
 * @author CMPS252
 * @version 6.0z Build 9000 Jan 26, 2020.
 */
public class FirstClass{
        /** Description of main(String[] args)
         *
         * @param args                   An array of string passed as
command line arguments
         * @return                       This method returns nothing
         */
        public static void main(String[] args) {
                System.out.println("Hello World");
        }
}
```

2- ☐ The format of the comments seen above, especially the tags (@author, version, etc..) is special to a program called `javadoc.exe` that takes as input annotated (tagged) source code and outputs HTML documentation: type `javadoc FirstClass.java↵`

3- ☐ Open the `index.html` file that was generated (double click on it or simply type `index.html` on the command prompt. Voila!