**Probal Chandra Dhar  
Advanced Programming Languages**

**Project 4  
User's Manual**

**Setup and Compilation**

1. Download and unzip the submission from eLearning on a windows machine.
2. The submission includes:

* VM.java
* Bytecode.java
* FuncMetaData.java
* Context.java
* Test.java
* readFromFile.java
* gcd.c
* gcd.exe
* GCDFunc.txt
* GCDEXEC.txt
* userManual-p4.docx

1. Environment: This program has been tested in the multi-platform lab (windows environment) and will run there.
2. Compiling. Please run the Test.java because there is the main() for this project.

**Running the program:** This submission includes the files listed above. Test.java has the main() for this project. When you run Test.java a file chooser will pop-up in the same directory you have the .java files. Please choose the files provided in the submission either GCDFunc.txt or GCDEXEC.txt. GCDFunc.txt includes the bytecode for the simple GCD program with the values of 1983 and 1530. Please change the line 59 & 60 if you want to change the number. It uses two functions to run the program. One is main & another one is gcdfunc. It has two literals also. After compiling the program with this file it will show the output to the console. The answer should be 3.

If you choose GCDEXEC.txt file from the file chooser it’ll exec() with literal pool from the file. Right no it’s this: c:\\test\\pdhar-p4\\gcd.exe 1983 1530. Please make sure you compile the gcd.c first so that it’ll create a .exe file. Sometimes IDEs make a.exe for the file you compiled. Please “Compile and link” if you’re using jGrasp. That will create gcd.exe. After a successful compilation it should also show the same result in the console. Output should be 3. Change the arguments in the literal pool if you want to calculate gcd of something else.

Please change the second parameter for the VM when you change between these programs. GCDFunc should use 2 and GCDEXEC should use 0 as there are no parameter.

I changed the argument of the VM constructor to create an object of type VM. Now, you need a add another String array for the literal pool, so that it can be passed to VM for EXEC.

*This Project needs Java 8 to compile. (Minimum Java 8)*

**User input:** After a successful compilation a jFileChooser window will pop-up in the screen and user must choose a bytecode file from the folder.

**Output:** all output goes to the console and every successful compilation will generate an output to the screen. Output will be the similar to this:

3

after GCDFunc/GCDEXEC