

CSC3320 System Level Programming Lab Assignment 6 - Part 2 - **Post Lab**

Due at 11:59 pm on Friday, Feb 26, 2021

Purpose: Learn the differences between writing a Bourne shell script and Java program. Learn how to use command argument in a Bourne Shell script. Learn how to compile and run Java and C programs in Unix terminal.

Part A:

Please complete the tasks in following table step by step and finish the questions below the table.

Step 1: Go to your home directory (`cd ~`) and create a new file named as **foo.sh** (**vi foo.sh**

```
#!/bin/bash
or nano foo.sh), then include following lines in your foo.sh.
#foo.sh in Part A of Lab 6 - Part 1
#

x=0 # initialization x = 0
i=1
while [ $i -le 3 ] # while(i<=3)
do
s=`expr $i \* $i` # s=i*i
x=`expr $s + $x`
i=`expr $i + 1` # i=i+1
done

echo x=$x
```

Step 2: Save your file and exit editor.

Step 3: Try following command to make simple.sh executable.

\$chmod a+x foo.sh

Step 4: Execute this file by invoking its name.

\$/foo.sh

*Note: when typing the shell script in your terminal, please be very careful of the **spaces**.*

Questions:

1) Attach a screenshot of the output in step 4.

```
[vdol10@gsuad.gsu.edu@snowball ~]$ vi foo.sh
[vdol10@gsuad.gsu.edu@snowball ~]$ chmod a+x foo.sh
[vdol10@gsuad.gsu.edu@snowball ~]$ ./foo.sh
x=14
[vdol10@gsuad.gsu.edu@snowball ~]$
```

2) Describe what does the shell script **foo.sh** do?

```
#!/bin/bash
```

```
#
```

```
#foo.sh in Part A off Lab 6 - Part 1
```

```
#
```

```
x=0 # initialization x = 0
```

```
i=1 # initialize i = 1 to use it as counter for loop
```

```
while [ $i -le 3 ] # while (i <= 3) run the loop until i <= 3; loop will run three times
```

```
do
```

```
s=`expr $i \* $i` # s = i * i; s = 1 in first interaction, s = 4 in second iteration, s = 9 in third iteration
```

```
x=`expr $s + $x` # x = s + x; x = 1 in first iteration, x = 5 in second iteration, x = 14 in third iteration
```

```
i=`expr $i + 1` # i = i + 1 # increment i by 1
```

```
done
```

```
echo x=$x # print x whose value is 14
```

Part B:

Step 1: Edit your *foo.sh* and change “-le 3” to “-le \$1”.

Step 2: When finished, save the *foo.sh* and exit editor. Then try executing it again by typing following command.

\$/foo.sh 5

Question:

Attach a screenshot of the output.

```
[vdol10@gsuad.gsu.edu@snowball ~]$ vi foo.sh
[vdol10@gsuad.gsu.edu@snowball ~]$ ./foo.sh 5
x=55
[vdol10@gsuad.gsu.edu@snowball ~]$
```

Part C:

Step 1: Edit your *foo.sh* in part B by making following modifications:

- ☐ Add two new lines below between line "**i=1**" and line "**while [\$i -le \$1]**"
echo please input a number
read num
- ☐ Change "**-le \$1**" to "**-le \$num**".

Step 2: When finished, save the *foo.sh* and exit editor. Then try executing it again by typing following command and **type 5** as the input of the number.
./foo.sh

Question:

Attach a screenshot of the output.

```
[vdol10@gsuad.gsu.edu@snowball ~]$ vi foo.sh
[vdol10@gsuad.gsu.edu@snowball ~]$ ./foo.sh
please input a number
5
x=55
[vdol10@gsuad.gsu.edu@snowball ~]$
```

Part D:

Write a Java program named **foo.java** to accomplish the same task as that in *foo.sh* of Part A.

Note: If you want to run your Java program in terminal,

- ☐ to compile *foo.java*, please try
\$javac foo.java
- ☐ To execute it, please try
\$java foo

Question:

Then put the source code of **foo.java** in your answer sheet.

```
[vdol10@gsuad.gsu.edu@snowball ~]$ vi foo.java
[vdol10@gsuad.gsu.edu@snowball ~]$ cat foo.java
public class foo {
    public static void main(String args[]) {
        int x = 0;
        int i = 1;
        int s;
        while (i <= 3) {
            s = i * i;
            x = s + x;
            i++;
        }
        System.out.println("x = " + x);
    }
}
[vdol10@gsuad.gsu.edu@snowball ~]$
```

```
[vdol10@gsuad.gsu.edu@snowball ~]$ javac foo.java
[vdol10@gsuad.gsu.edu@snowball ~]$ java foo
x = 14
[vdol10@gsuad.gsu.edu@snowball ~]$
```

Part E:

Create and run Kernighan and Ritchie's famous "hello,world" program. **Step 1:** Go to your home directory (cd ~) and create a new file named as **hello.c** (vi hello.c

or nano hello.c), then include following lines in your **hello.c** .

```
#include <stdio.h>

int main(void)
{
    printf("Hello,world\n");
    return 0;
}
```

Step 2: Save your file and exit editor.

Step 3: Compile and link the hello.c program by following command.

\$cc hello.c

***Note:** after this command, a default executable program named as "a.out" will be generated in current directory if there are no errors with your C program. You can use **ls** to check the existence of a.out .*

Step 4: Run the executable program **a.out**

\$/a.out

```
[vdol10@gsuad.gsu.edu@snowball ~]$ vi hello.c
[vdol10@gsuad.gsu.edu@snowball ~]$ cat hello.c
# include <stdio.h>

int main(void) {
printf("Hello,world\n");
return 0;
}
[vdol10@gsuad.gsu.edu@snowball ~]$
```

Questions:

1) Attach a screenshot of the output in step 4.

```
[vdol10@gsuad.gsu.edu@snowball ~]$ cc hello.c
[vdol10@gsuad.gsu.edu@snowball ~]$ ls
a.out          file1.txt      file4.pdf      foo.sh         homeworks     pdffiles       Result         txtfiles.tar.gz
checkError.sh  file2.pdf      foo.class      hello.c        Lab3          pdffiles.tar.gz simple.sh
csc3320        file3.txt      foo.java       hello.sh       Lab4          public         txtfiles
[vdol10@gsuad.gsu.edu@snowball ~]$ ./a.out
Hello,world
[vdol10@gsuad.gsu.edu@snowball ~]$
```

2) Try following command to compile and link **hello.c** again. And tell what new file is generated after this command?

\$cc -o hello hello.c

hello is the new file after using this command

```
[vdol10@gsuad.gsu.edu@snowball ~]$ cc -o hello hello.c
[vdol10@gsuad.gsu.edu@snowball ~]$ ls
a.out          file1.txt  file4.pdf  foo.sh      hello.sh    Lab4        public      txtfiles
checkError.sh  file2.pdf  foo.class  hello       homeworks  pdffiles    Result      txtfiles.tar.gz
csc3320        file3.txt  foo.java   hello.c     Lab3        pdffiles.tar.gz  simple.sh
```

3) Try the command below and attach a screenshot of the output.

\$/hello

```
[vdol10@gsuad.gsu.edu@snowball ~]$ ./hello
Hello,world
[vdol10@gsuad.gsu.edu@snowball ~]$
```

4) Now write a new C program named as **myName.c** based on **hello.c**. In this program, print out your first name and last name instead of "Hello,world". For example, the output could be "My name is Yuan Long".

Execute your **myName.c** and attach a screenshot of the output. Then write the source code

of **myName.c** in your answer sheet and upload your file **myName.c** to classroom.

```
[vdol10@gsuad.gsu.edu@snowball ~]$ vi myName.c
[vdol10@gsuad.gsu.edu@snowball ~]$ cat myName.c
# include <stdio.h>

int main(void) {
printf("My name is Vivian Do\n");
return 0;
}
[vdol10@gsuad.gsu.edu@snowball ~]$
```

```
[vdol10@gsuad.gsu.edu@snowball ~]$ cc myName.c
[vdol10@gsuad.gsu.edu@snowball ~]$ ls
a.out          file1.txt  file4.pdf  foo.sh      hello.sh    Lab4        pdffiles.tar.gz  simple.sh
checkError.sh  file2.pdf  foo.class  hello       homeworks  myName.c    public          txtfiles
csc3320        file3.txt  foo.java   hello.c     Lab3        pdffiles    Result          txtfiles.tar.gz
[vdol10@gsuad.gsu.edu@snowball ~]$ ./a.out
My name is Vivian Do
[vdol10@gsuad.gsu.edu@snowball ~]$
```

Submission:

Note: Please follow the instructions below step by step, and then write a report by answering the questions and upload the report (named as Lab6_FirstNameLastName.pdf or Lab6_FirstNameLastName.doc) to Google Classroom, under the rubric Lab 6 Out-of-lab Assignment.

Please add the lab assignment NUMBER and your NAME at the top of your file sheet.