CSE 15L Winter 2017
Exam Practice Questions

Example True/False questions:

1. When debugging, it is a best practice to make one change at a time and keep a record of the charges. [T]
2. Shell scripts are not executable by default. [T]
3. In a shell script, line starting with #! are comments. [F]
4. In Unix, the command line arguments passed to a shell script are stored in positional parameters. [T]
5. A shell acts as an interface between the user and the system. [T]
6. As a command interpreter, the shell sends the commands to the kernel for execution. [T]
7. A directory is a specialized form of file that maintains a list of all files in it. [T]
8. Unit testing is the testing of the whole class as a unit. [F - method]
9. JUnit is a unit testing framework for the Java Programming Language. [T]
10. Debugging is not algorithmic. [T]
11. In test-driven development, tests are written before the software to be tested is written. [T]
12. It is usually feasible to test all possible values of arguments to a method. [F]
13. stdout should be used for unusual output from your program, such as error reporting. [F - stderr]
14. In a Makefile, every target line must begin with a tab. [F – action line]
15. In debugging, reproducing the problem comes before fixing the problem. [T]
16. If all your unit tests pass, that means the software being tested has no bugs. [F]
17. A software version management repository holds a master copy of only text/html code. [F – other artifacts as well]
18. Making changes to a software repository means editing, adding and deleting files. [T]
19. IDE stands for "integrated development environment". [T]
20. Eclipse does not allow you to have multiple workspaces. [F – you can have multiple workspaces]
21. In test-driven development, you must understand the specification of what the software you are testing is supposed to do. [T]
22. In Unix, executing programs have unique process identifiers. [T]
23. A Unix pipe is a way to send the output of one command to the input of another. [T]
24. As a login script, .profile runs each time a Unix shell is started. [T]
25. The Makefile contains rules which tell make what to do. [T]
26. If a target in a Makefile names a file that is older than the files it depends on, then the actions in that rule will not be performed. [F]

Other example questions:

1. List three components of a good development environment for a mid-sized project team.

[In lecture slides.]

2. List two cons of test-driven development.

[In lecture slides.]

3. How is the command

```
$cat file2
```

different from

`$cat >file2` and `$cat >> file2`

?

[cat file2 displays the contents of file2, where as > and >> redirect the input to file2. > overwrites and >> appends to the contents of the file.]

4. What is redirection in unix? Provide an example command that uses redirection.

[Directing the flow of data to the file or from the file for input or output. Example : ls > wc]

5. What Unix command is used for changing file access permissions?

[chmod]

6. What does the command "$ls | wc –l > file1" do?

[ls becomes the input to wc which counts the number of lines it receives as input and instead of displaying this count , the value is stored in file1.]

7. What is a pipe and give an example?

[A pipe is two or more commands separated by pipe char '|'. That tells the shell to arrange for the output of the preceding command to be passed as input to the following command.

Example : ls -l | pr

The output for a command ls is the standard input of pr.

When a sequence of commands are combined using pipe, then it is called pipeline.]

8. One can change the access permissions on a file called 'temp.txt' to readable, writeable and executable only by the file owner.

[chmod 700 temp.txt]

9. Construct the pipe to execute the following job.

"Output of ls should be displayed on the screen and from this output the lines containing the word 'poem' should be counted and the count should be stored in a file."

[ls | grep poem | wc -l > out.txt]

10. The command

```
$ cat par.3 par.4 par.5 >> report
```

a. displays contents of the files par.3, par.4 and par.5 on the command line and overwrites the file report with the output
b. appends the file report with the contents of par.3, par.4 and par.5
c. displays contents of the files par.3, par.4 and par.5 on the command line and appends the file report with the output
d. overwrites the file report with the contents of the file par.5

11. The positional parameter $0 refers to:

   a. Total number of arguments
   b. Command or the script name
   c. First argument respectively.
   d. All the command line arguments starting from $1.

12. Define a shell array called CLASS with elements “Altintas”, “CSE15L”, “Fall 2013” in the given order. Display the second element with a Unix command.

[
```
CLASS[0]=“Altintas”
CLASS[1]=“CSE15L”
CLASS[2]=“Fall 2013”
echo "Second element: ${CLASS[1]}"
```
]

13. Complete the following sentence. “_____[pwd]__________ command displays the current working directory in Unix shell.”

14. List three tools for diagnostic output.

[In lecture slides.]

15. List three pieces of information provided by Java logging API.

[In lecture slides.]

16. Name three steps in debugging and put them in order.

[In lecture slides.]

17. Fill in the blank: The binary output from hprof can be opened using _[jhat]___.

18. List two things a bug reporter should read or check to report a good bug.

[In lecture slides.]

19. Write a Unix commands that displays the contents of folder “Q1” in long form including the directory entries whose names begin with a dot (.).

   ls –la Q1

20. List three common Unix shells.

   bash, ksh, tcsh, …

21. Write a command to make the file “Q3.txt” in the current directory read-only.

   chmod 444 Q3.txt

22. Absolute path in Unix paths start at:

   a. users
   b. bin
   c. home
   d. root

23. Relative to the working directory, *dot dot* (..) refers to:

   a. current working directory
   b. home

c. root

d. a level above

24. List the names of the two files that a shell automatically opens when it starts up.

[.bashrc, .profile]

25. Write a command that redirects the contents of standard input to an output file called 'temp.txt'.

[cat > temp.txt]

26. Write a filtering command using Unix pipes to search for occurrences of the word 'time' in the manual page for Unix command called "time".

[man time | grep time]

27. What is the function of the Unix environment variable "PATH".

[The PATH environment variable is a colon-delimited list of directories that your shell searches through when you enter a command.]

28. List two cons and two pros of unit testing. (in lecture slides)

29. Which file does `make` look for in the current working directory? (Makefile)

30. For this problem, consider this Makefile:

```
A.class: A.java
    javac A.java
B.class: A.class B.java
    javac B.java
C.class: B.class A.class C.java
    javac C.java
all: C.class B.class A.class
```

a. Suppose ls -l produces this output:

```
-rw-rw-r-- 1 u1 lec 150 May 7 13:16 A.java
-rw-rw-r-- 1 u1 lec 81  May 7 13:25 B.java
-rw-rw-r-- 1 u1 lec 189 May 7 13:20 C.java
-rw-rw-r-- 1 u1 lec 168 May 7 13:00 Makefile
```

Now running "make all" will cause what command(s) to be executed? List them.

```
javac A.java
```

```
        javac B.java
        javac C.java
```

b. Suppose ls -l produces this output:

```
        -rw-rw-r-- 1 u1 lec 356 May 7 13:50 A.class
        -rw-rw-r-- 1 u1 lec 150 May 7 13:16 A.java
        -rw-rw-r-- 1 u1 lec 251 May 7 13:50 B.class
        -rw-rw-r-- 1 u1 lec 69  May 7 13:25 B.java
        -rw-rw-r-- 1 u1 lec 532 May 7 13:50 C.class
        -rw-rw-r-- 1 u1 lec 189 May 7 13:55 C.java
        -rw-rw-r-- 1 u1 lec 168 May 7 13:00 Makefile
```

Now running make all will cause what command(s) to be executed? List them.

```
javac C.java
```

31. Complete the following sentence to explain how make interprets the Makefile rules. “To make the ____*target*__________ , first make all its _____*dependencies*_______________ , then perform all the ____*actions*________.”

32. List two problems that can arise in non-version controlled software when there is only one individual involved in the software development process. (in lecture slides)

33. Put the following steps in order within an version control workflow: makes changes, update, commit, resolve conflicts. (You can use a step more then once if needed.) (in lecture slides)

34. Name three steps in debugging and put them in order. (in lecture slides)

35. Fill in the blank: A string enclosed in back quotes (``) in a Unix shell script is treated as a ____*command*_________ and the shell attempts to execute it.

36. Complete the following shell script.

```
        #!/bin/sh
        echo Total number of inputs: ____$#_____
        echo First input: ____$1_____
        echo Second input: _____$2____
```

37. Square brackets ([ ]) in a Unix shell script is an alias for the ____*test* ___ command.

38. Answer the following questions about the Unix system:
    a. What is the purpose of ‘2>’ in Unix?
    b. How do you redirect the output of a script called “***myscript.sh***” to a file called “***out.txt***”?
    c. How do you search for the pattern ***“history”*** in a file called “***text.txt***”?

39. Fill in the blanks in the following shell script to write a while loop that displays “Hi” followed by the input string the user enters until the user enters an empty string.

```
#!/bin/bash
yname="foo"
while ["$yname" ______  "" ]
___________
        echo -n "Enter your name : "
        read yname
        if _______"$yname" != ""
        then
                echo "Hi _____________"
        ____________
____________
```

40.
    a. Explain the following commands in Git:

`commit:`____

`push:`____

add: ____

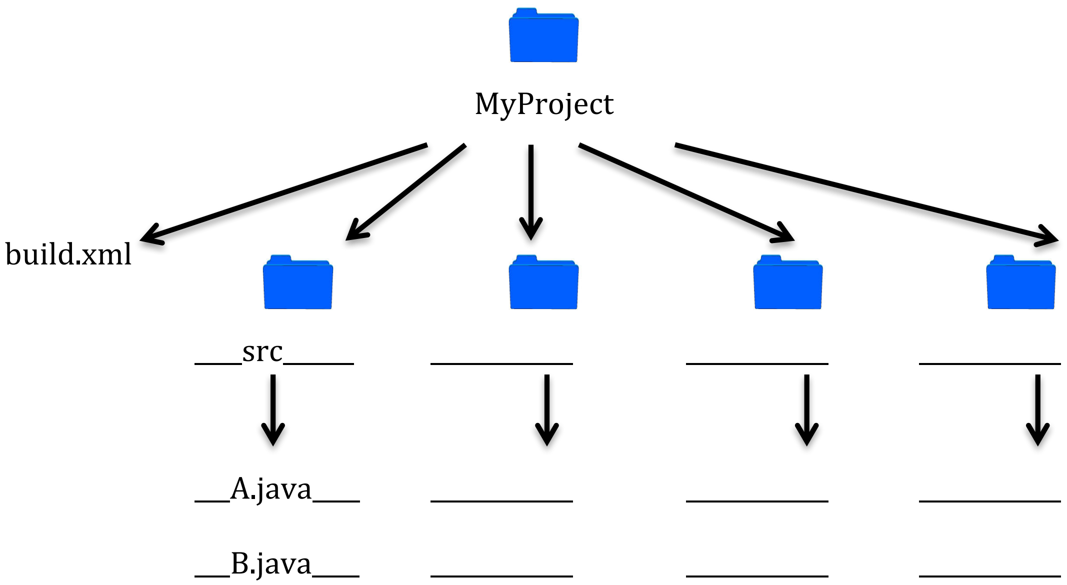
b. List the execution order of the commands in 6.a.

c. What is ”`git init`“ is used for?

41. Please use the following build.xml file in Ant to answer the following two questions.

```
<project name="MyProject" default="doc" basedir=".">

  <property name="src" location="src"/>
  <property name="build" location="build"/>
  <property name="doc"  location="doc"/>

  <target name="init">
    <mkdir dir="${build}"/>
    <mkdir dir="${doc}"/>
  </target>

  <target name="compile" depends="init" description="compile the source" >
    <javac srcdir="${src}" destdir="${build}"/>
  </target>

  <target name="doc" depends="compile" description="generate
documentation">
    <javadoc sourcepath="${src}" destdir="${doc}"/>
  </target>

  <target name="clean" description="clean up" >
    <delete dir="${build}"/>
    <delete dir="${doc}"/>
  </target>
</project>
```

a. (6 pts) Draw the project directory structure after you run "ant compile". (Hint: Not all the blank lines need to be filled in. Leave them blank if you think there are extras.)



b. (2 pts) What does the doc target accomplish?

c. (2 pts) Which target will run if you only run "ant" on command for this build file?

d. (3 pts) What targets does the doc target depend on?