CSCI 2961 – Introduction to Open Source

First Quiz, Open Book, Open Notes, Closed Internet
Please Answer All Questions. Total number of points possible 70
November 16, 2015

NAME: Sample Solution.

1. Please Answer the following Multiple Choice Questions: [*30 points*] Circle the answer with a circle ◯ on the answer.

   (a) Linux kernel was developed by
      i. Richard Stallman
      ii. Sergei Brin and Larry Page (Google Founders)
      iii. Steve Jobs (Apple)
      (iv) Linus Torvalds

   (b) Which one of them is NOT an example of high quality Open Source Development?
      i. Apache
      ii. LaTeX
      iii. Mozilla/Firefox
      (iv) Adobe Acrobat (pdf manipulator)

   (c) Which one is NOT a required freedom for Free Software Development?
      (i) Freedom to run the program but not for illegal activities
      ii. Freedom to improve the program and releasing your source code with improvements
      iii. Freedom to distribute copies
      iv. Freedom to study how the program works and change it to make it do what you wish.

   (d) If you want to start an open source project what is the least needed one.
      i. Open Source Repository
      ii. Open Source License
      iii. Development, Testing and Documentation Plan
      (iv.) Marketing Division

   (e) In Open Source Software Development, which is the most essential?
      (i.) Smart Data Structure, Dumb Code
      ii. Dumb Data Structure, Smart Code
      iii. DO not release your code early and Often
      iv. Treat your community with contempt.

   (f) Which is a true indication of an Open Source Software?
      i. Any source code in a website
      ii. Any source code in github
      (iii.) Any source code with a OSI approved license
      iv. Your friends code

   (g) Which of the following is NOT a part of Open Source Definition

i. Derived works
ii. Distribution of License
iii. License must restrict other software
iv. Integrity of Author's source code.

(h) Verification is:

i. Checking that we are building the right system
ii. Checking that we are building the system right
iii. Performed by an independent test team
iv. Making sure that it is what the user really wants

(i) A regression test:

i. Will always be automated
ii. Will help ensure unchanged areas of the software have not been affected
iii. Will help ensure changed areas of the software have not been affected
iv. Can only be run during user acceptance testing

(j) If an expected result is not specified then:

i. We cannot run the test
ii. It may be difficult to repeat the test
iii. It may be difficult to determine if the test has passed or failed
iv. We cannot automate the user inputs

(k) Which of the following could be a reason for a failure

```
1) Testing fault
2) Software fault
3) Design fault
4) Environment Fault
5) Documentation Fault
```

i. 2 is a valid reason; 1,3,4 and 5 are not
ii. 1,2,3,4 are valid reasons; 5 is not
iii. 1,2,3 are valid reasons; 4 and 5 are not
iv. All of them are valid reasons for failure

(l) The process starting with the terminal modules is called

i. Top-down integration
ii. Bottom-up integration
iii. None of the above
iv. Module integration

(m) How much testing is enough

i. This question is impossible to answer

ii. The answer depends on the risks for your industry, contract and special requirements

iii. The answer depends on the maturity of your developers

iv. The answer should be standardized for the software development industry

(n) Path Coverage method comes under which testing method.

i. White box

ii. Black box

iii. Green box

iv. Yellow box

(o) To test a function, the programmer has to write a ————————, which calls the function and passes it test data.

i. Stub

ii. Driver

iii. Proxy

iv. None of the above

2. **Testing** The following python Code is given an example of a quicksort code. Please comment on this code for efficiency. Please add a brief description of what the code does. One test case is shown. Provide at least three cases with python unittest code. (we have given examples for collinear, gcd and for bsearch) [*10 points*]

```
def quick(lst):
 l1 = [x for x in lst[1:] if x <= lst[0]]
 l2 = [y for y in lst if y > lst[0]]
 if len(lst) == 0:
  return lst
 elif len(lst) == 1:
  return lst
 else:
  return quick(l1) + [lst[0]] + quick(l2)

## Sample Test
''' Sample Test for quicksort given above - uses module random '''
import random
my_randoms = random.sample(xrange(100), 10)
print(quick(my_randoms))
```

1. Takes the first element of the list to split the array/list [smaller than the first element, orequal to first element, greater than the first element]
2. Recursively calls the left and right sublists.
3. very inefficient Worst case $O(n^2)$ - choice as a first element is a bad choice; creates additional lists (instead of a single list). (e.g. already sorted array, array in decreasing order)

For ~~test~~ unittest See the next sheet

```
import unittest
import random
from qsort import quick

class TestUM(unittest.TestCase):

    def setUp(self):
        pass
    def test_0(self):
        lst1=[]
        self.assertEqual(quick(lst1), [])

    def test_char(self):
        lst1=['r', 'p','i']
        self.assertEqual(quick(lst1), ['i','p','r'])
    def test_big_array(self):
        lst1 = [random.randint(1,100000) for _ in xrange(100)]
        lst2=lst1
        lst2.sort()
        self.assertEqual(quick(lst1),lst2)

if __name__ == '__main__':
    unittest.main()
```

If this is changed to 100,000 it gives error because of the inefficiencies pointed out.

3. **Git Commands** [*Total 10 points*]

(a) Write down the git commands you will do to clone a repository git@github.com:mskmoorthy/open-source-class.git , add a new file called <myname-mytest>.md (with the content having bullet item lists of your name and your year of graduation) and push to that directory.

What will happen after you push. [*3 Points*]

1. git clone git@github.com:mskmoorthy/open-source-class.git
1.5 cd open-source-class
2. vim myname-mytest.md
insert
- mskmoorthy
- 1900
esc :wq
3. git add myname-mytest.md
4. git commit -m 'Added a file'
5. git push

(b) Execute the git commands to go from Figure 1 to Figure 2. In Figure 1 c1 is the Master(current head pointer) and Figure 2 C3 is the master (current head pointer) [*2 Points*]

(c) Please list the commands to do the following: Create a fork of a remote repository git@github.com:mskmoorthy/open-source-class.git. Get a copy of the repository in your local machine. Create a branch bugfix. In that branch fix all the spelling mistakes. Push that branch to your github repository. Give a pull request. [*5 Points*]

(Lab problem)

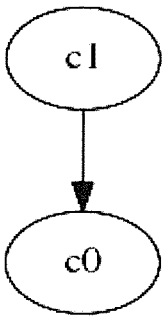
b) git commit
git commit

6

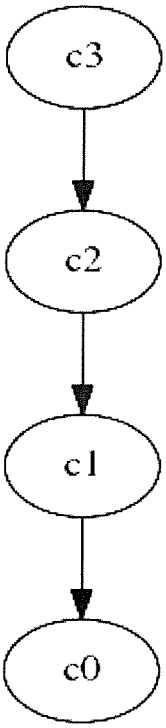
c) 1. go to github.com and fork the repository
2. git clone ....
3. git branch bugfix
4. git checkout bugFix

6

5. cd ....
6. git add .
7. git commit -m "fixed bugs"
8. git push



(a) Graph 1



(b) Graph 2

4. **Build Systems** [*Total 10 points*]

(a) The command **gcc <filename.c>** does more than compiling. Provide at least two other operations that gcc provides [*2 points*]

1. Creates an archive file.
2. To create and update "static library" files that the linker uses.

1. Assembles the compiled files.
2. Does macro expansion
3. Links all libraries and create an executable file.

(b) what does the command **ar** do? Explain the difference between static library and shared library? [*2 points*]

1. creates an archive file (from a group of files)
   - (tar, shar comes from ar)
2. To create and update "static library" files that the linker uses

Static library - use to link executables - all symbols with extern exposed publicly

shared library - link executables to it at run time by linker searched

(c) Explain the advantages of a build system **make** [*2 points*]

1. Compiles files only that have been changed since the last make command.
2. Uses the dependen

(d) List Four features of the command **cmake**. What is the input to the command **cmake** [*4 points*]

1. Use platform-and-tool-Specific commands
2. Handle implicit dependencies automatically
3. Provide rules for "install" and "clean" operations
4. Display the progress of each operation with percentage
5. Pristine & Source with out of source builds

5. **Documentation and Misc Topics** [*Total points 10*]

(a) What is the difference between markdown and markup languages? How is markdown different from HTML? [*2 points*]

markdown languages are also marked up languages - a stripped version for security reasons. markdown languages have much simpler syntax Cannot include scripts and css in markdown languages.

(b) Give at least two reasons (advantages) of having a good documentation [*2 points*]

1. Ease of maintenance and usability

2. Ease of extending the software for other uses.

(c) Solve the following 2x2 crossword puzzle where the clues are given as regular expressions *6 points*] (**Watch out for meta characters and subtle regular expression features**)

Across Clues:

1. [*]+ (square bracket star square bracket +)

2. /+ (slash +)

Down Clues:

1. .?.+ (dot question mark dot+)
2. . + (dot+)