Open Source Software — CSCI-4966-01 — Spring 2019
Test 1
February 22, 2018

Name: _____

RCS ID: | | | | | | | | | @rpi.edu |

RIN#: _____

Honor pledge: On my honor I have neither given nor received aid on this exam.

Please sign here to indicate that you agree with the honor pledge: _____

Instructions:

- Clearly print your name, RCS ID (in all caps.) and your RIN at the top of your exam.
- This test is open book, open notes and open computer. You **m**ay not use the internet. Please turn off your wifi.
- There are **8 questions** on this test worth a total of **100 points**.

Question	Score	Possible
1		15
2		13
3		12
4		15
5		10
6		15
7		6
8		14
Total		100

1. The Open Source Initiative defines 10 characteristics of open source software. List 5 along with a **brief** explanation of what they mean: (15 pts)

   1 -

   2 -

   3 -

   4 -

   5 -

2. Consider the scenario where a company is using open source software to produce applications that they then sell.

   (a) What is the impact (license, ability to distribute, etc.) on their product if the open source software they used is licensed under: (9 pts)

      i. GPL:

      ii. LGPL:

      iii. MIT:

(b) Which license (in the open source software they are using) serves the company best if their business plan is to (4 pts):

i. Sell unique software that they developed which uses the Open Source Software

ii. Provide support services around the Open Source Software they are using without creating new applications or code

Make sure you support your answer.

3. For each question below, circle the best answer (12 pts)

(a) Which command changes you repository to a new (existing) branch?

i. git branch
ii. git status
iii. git checkout newbranch
iv. git log

(b) Literate programming:

i. Is well structured code with minimal comments
ii. Cannot express complicated algorithms
iii. Mixes code and comments in an easily human readable format
iv. Is a failed development methodology

(c) Open or Free software:

i. Cannot be used for a commercial purpose
ii. Can be redistributed either for free or for a fee
iii. Can be sold, but only for the nominal cost of the media used to store it
iv. Must be maintained by unpaid volunteers

(d) It is a good idea when selecting an open source license to:

i. Go to an authority such as the OSI and pick an approved license
ii. Create a new license from scratch because it is unlikely that an existing license would meet your needs
iii. Just place the code in a public repository. Easily available code is the same as open
iv. Take an existing, approved license and modify to better represent your unique personality

4. Give a sequence of git commands to accomplish the following (you can assume that you are always working on the "master" branch") (15 pts):

   (a) Create a local copy of your repoository by cloning from "https://www.mypublicrepository/public.git". This will be your origin.

   (b) Create an "upstream" remote to point to "https://www.everyonespublicrepository/public.git"

   (c) Assume you have a new file "foo.txt" in your local directory. Add this file to your repository.

   (d) Send your changes to the public repository

   (e) Assume someone else makes changes in the "upstream". Add the changes in the public repository into your local version.

   Write git commands below:

5. Write markdown to duplicate the document below. You can assume the photo name is "photo.jpg" (10 pts):

Test File - Biggest Header

Next Smallest Header

1. Enumerated list
2. Of multiple lines
3. Just something else to type

A clickable link to Google (http://google.com) here: Google link

A picture from file photo.jpg in the current directory:

Write Markdown commands below:

6. Assume you have 3 source files: a main file "prog.c" and two additional files"f1.c", and "f2.c" containing code that "prog" depends upon. Write a Makefile that explicitly creates object files for all 3 sources, create a static library from the "f1.c" and "f2.c" object files, and then create an executable named "prog.exe". Make sure your Makefile contains appropriate "all" and "clean" targets. (15 pts):

Write your Makefile below:

7. Repeat the previous exercise for CMake. (6 pts): Write your CMakeLists.txt file below:

8. Consider the following code binary search code in file binary.py:

```
def binary_search( x, L):
    low = 0
    high = len(L)
    while low != high:
        mid = (low+high)//2
        if x >= L[mid]:
            low = mid+1
        else:
            high = mid
    return low
```

The code contains an error.

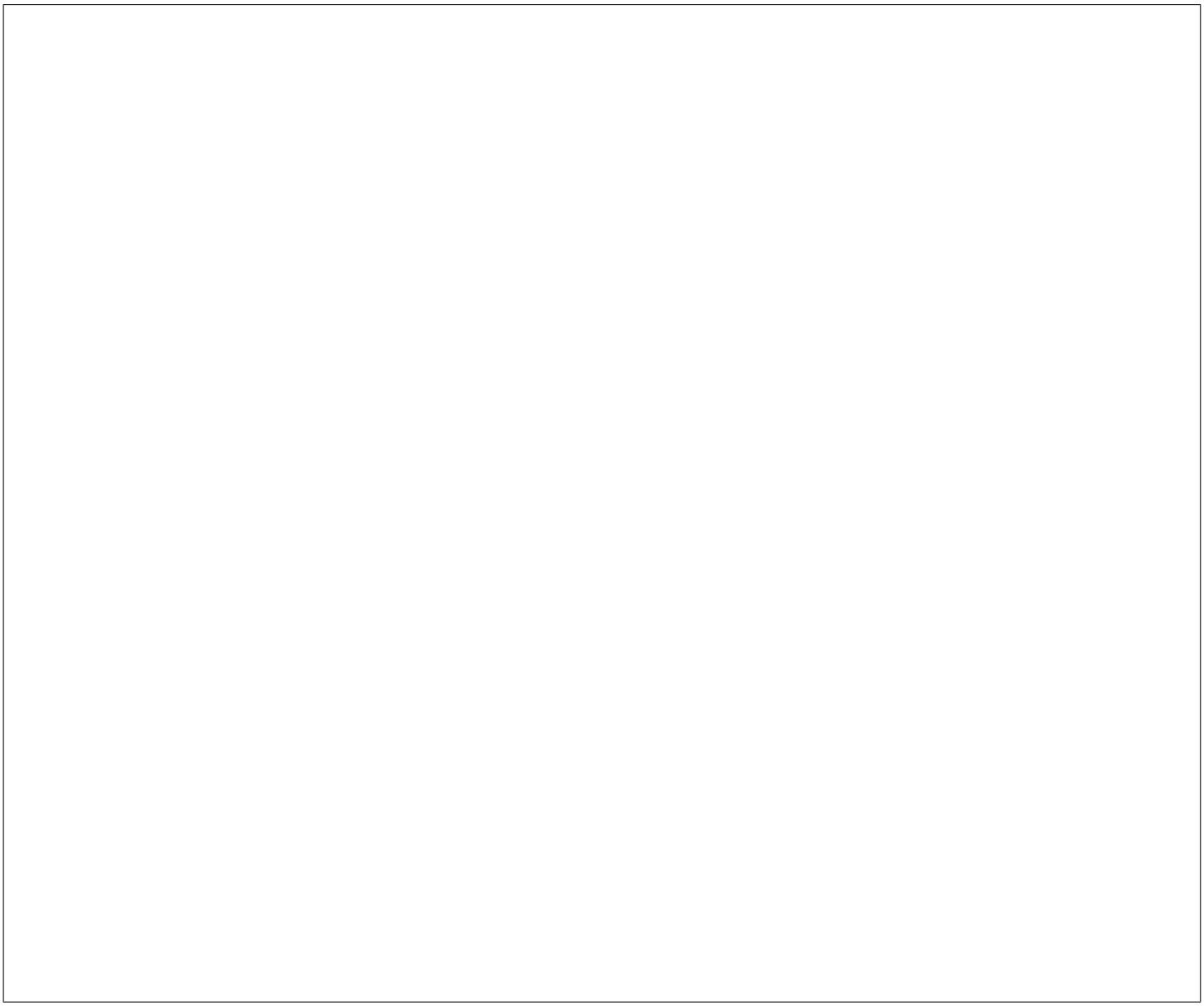
(a) Find and correct the error. Write your correction alongside the code above. You may use any tools on your computer to help you find the error. (2 pts)

(b) Now, on the next page, write a sequence of tests in file binary_unit_test.py to ensure that the algorithm is correct. For full credit you will need to cover at least 4 different use cases. We give you the start and the end of the test file (12)

```
'''
Test binary.py with unittest
To run tests:
python binary_unit_test.py
'''
import unittest
from binary import binary_search

class TestBinaryPy(unittest.TestCase):

    def setUp(self):
        pass
```



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