

SENG 265 – Software Development Methods
Fall 2023
Midterm Exam: Wednesday, 18 October 2023
Instructor: Roberto A. Bittencourt

Students must check the number of pages in this examination paper before beginning to write, and report any discrepancy immediately.

- **All answers are to be written on this exam paper.**
- We will not answer questions during the exam. If you feel there is an error or ambiguity, write your assumption and answer the question based on that assumption.
- The exam is closed book. No books or notes are permitted.
- When answering questions, please do not detach any exam pages!
- ***Electronic devices are not permitted.*** Cellphones must be turned off.
- The marks assigned to each section are printed within parentheses. Partial marks are available for questions in Sections B and C.
- There are thirteen (10) printed pages in this document, including this cover page.
- Last page is left blank for scratch work. **Answers on the that page will not be graded.**
- We strongly recommend you read the entire exam through from beginning to end before starting on your answers.
- **Please have your UVic ID card available on the desk for inspection.**

Section A (30 marks): For each question in this section, place an X **beside all answers that apply**. Each question is worth 3 (two) marks. *Partial marks are not given for incomplete answers.*

Question 1: By stating that *git* is a *DISTRIBUTED* version-control system, we are saying:

- ___ various repositories may be used by clients to deal with a single software project.
- ___ if the remote repository breaks down, it is not possible to recover commit history.
- ___ each user with access to a remote repository may keep a copy of the repository history at its own computer.
- ___ It permits concurrent read and write access of remote repositories to users who have read- and write-access rights to those repositories.
- ___ None of the above.

Question 2: *git clone ssh://billg@git.seng.uvic.ca/seng265/billg*

- ___ copies a full remote repository to a local repository.
- ___ gives the local repository access to only the latest commit made to that remote repository.
- ___ Can only be performed once for the *billg* project.
- ___ Includes the full commit history of the project stored at that remote repository.
- ___ None of the above.

Question 3: The *git commit* command:

- ___ is the same as the *git push* command.
- ___ sends all the staged files in the staging area to a remote repository.
- ___ sends all the staged files in the staging area to the remote central server.
- ___ reverses the effect of the most recent *git clone* command.
- ___ None of the above.

Question 4: If a directory has the permissions *drwxr-xr--*, then this means:

- ☐ anyone with access to this system is able to enter the directory.
- ☐ anyone with access to this system is able to list the directory contents.
- ☐ the owner of the directory cannot remove the execute permission of the group the directory belongs to.
- ☐ only the owner is able to delete the directory.
- ☐ None of the above.

Question 5: The UNIX *mv* command:

- ☐ Can be used to create symbolic links to an existing file as a type of “shortcut”.
- ☐ Can be used to rename a file.
- ☐ Can be used to move a file to another directory.
- ☐ Can be used to delete directories if used with the *-r* recursive option.
- ☐ None of the above.

Question 6: A pipe (“|”) that may be constructed using the bash shell:

- ☐ connects the *stdout* stream of one command with the *stdin* stream of the next command.
- ☐ changes all the file streams in the second command in the pipe to the *stdout* stream.
- ☐ connects the *stdout* stream of one command with the *stdout* stream of the next command.
- ☐ redirects the standard output of the command after the pipe to a file.
- ☐ None of the above.

Question 7: The *while*-loop in C:

- ___ Is a bottom-tested loop just like the *for* statement.
- ___ May contain assignment commands inside its parentheses (round brackets).
- ___ May contain statements that dereference pointers.
- ___ Will execute the commands inside its body at least once.
- ___ None of the above.

Question 8: The C string function *strncpy(char *dest, char *src, int len)*:

- ___ Copies at most *len* characters from *dest* to *src*.
- ___ Always copies exactly *len* characters from *src* to *dest*.
- ___ Is considered less safe to use than *strcpy*.
- ___ Raises a *StringLengthException* if *src* string is longer than *dest* string.
- ___ None of the above.

Question 9: When using Python 3 lists, _____ .

- ___ a copy of the list contents is made as the result of a statement such as `some_other_list = somelist[:]`
- ___ a negative lookup (such as `somelist[-3]`) refers to list values positioned relative to the end of the list
- ___ given some list, we can refer all the values from the second item to the fifth item using `somelist[1:5]`
- ___ immutable types such as tuples cannot be added as list elements
- ___ None of the above.

Question 10: A Python 3 function:

- ___ always declares the type of each its parameters when there are parameters.
- ___ may be passed as a function parameter.
- ___ may be overloaded, i.e., may allow other functions with the same name to be defined with a different number of parameters and a different body.
- ___ always has a return value, even if no return statement is called inside the function.
- ___ None of the above.

Section B (30 marks): Two questions, each one worth 15 (ten) marks.

Question 11: (15 marks). Consider the following Python 3 code:

```
1 def f(q, w):
2     return q(w)
3
4 def f2(n):
5     for i in range(n):
6         # using end='' does not add a newline at print end
7         print('%d ' % i , end='')
8     print()
9
10 def f1(x):
11     for i in range(1, x + 1):
12         f(f2, i)
13
14 def main():
15     number = 5
16     f(f1, number)
17
18 main()
```

Trace the code's output below, as it would appear in the console screen.

Question 12: (15 marks). Consider the following C program (with line numbers provided in the left-hand margin for your convenience):

```

1 int main() {
2     int *p, *q, *r, *s, *t, *u;
4     int a=0, b=0, c=0, d=5, e=0, f=0;
5
5     p = &b;
6     q = &d;
7     s = &a;
8     r = &c;
8     *s = 10;
9     *r = a + *q;
10    t = s;
11    *p = *r;
12    u = &c;
13    *u = (*p + *r) * *s;
14    e = *q - b;
15    /* what is the state of the memory at this point
16       before the main function ends? */
17 }

```

Suppose that the image below represents the memory of the program above. Fill out the content of the memory cells before the main function ends (after line 14).

| <i>address</i> | <i>Memory</i> | <i>variable</i> |
|----------------|---------------|-----------------|
| 10500 | 10528 | p |
| 10504 | | q |
| 10508 | | r |
| 10512 | | s |
| 10516 | | t |
| 10520 | | u |
| 10524 | | a |
| 10528 | | b |
| 10532 | | c |
| 10536 | | d |
| 10540 | | e |
| 10544 | 0 | f |

Section C (40 marks): One question

Question 12

Write a C function with the following signature:

```
void remove_duplicate_characters(char *source, char *destination)
```

This function accepts two strings `source` and `destination` as parameters and modifies the `destination` string so that it contains only the unique chars in `source`. The original `source` string must not be modified. *Memory for the resulting string must NOT be dynamically allocated.*

The characters in the `destination` string must be ordered in the same order they originally appear in the `destination` string. For example, if the following commands are run:

```
char source[100] = "house of the rising sun";
char destination[100];
remove_duplicate_characters(char *source, char *destination);
printf("%s\n", destination);
```

the console should show:

```
house ftring
```

Do not modify the contents of the `source` string passed to your function as a parameter.

Some marks will be given for the quality of your solution.

(The next blank page of this exam may be used for your solution.)

(This page intentionally left blank for scratching.)