University of Victoria

Examinations April 2007

SENG265 S01, S02 – Software Development Methods

Name: ______________________________ Student No. __________

Instructor: Michael Zastre Section: _____________

Duration: 3 Hours

To be answered on the paper.

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

This question paper has 13 pages.

- You must obtain permission from an invigilator to leave the examination room temporarily.
- No cell phones are permitted.
- This is a closed-book, closed-notes exam.
- Write legibly.

Question	Mark
1	
2	
3	
4	
5	
6	
7	
8	
Total (133 max)	

Question 1: Versioning Systems (9 marks)

You have applied to be a co-op programmer for a small company in Victoria. During the interview, the interviewer "plays dumb" and pretends her company has no versioning systems (such as CVS) at their company. She is trying to determine your technical knowledge by checking your reactions. What do you say in response to the following questions and comments?

a) "Programmers at our company have seen advertisements for these versioning-control systems, but say it is a waste of time as they keep versions in their directories. Do you think it is a waste of time?"

b) "I don't quite see what the problem is with conflicts when two programmers edit the same code. Why not let the computer take care of all the conflicts?"

c) "We don't like CVS because it forces you to work at the office when obtaining code from the repository."

Question 2: Testing (12 marks)

a) When is *white-box testing* preferred over *black-box testing*?

b) What is meant by the term *unit test*, and what purpose does such a test serve in the development process?

c) What is the meaning of *test coverage*?

Question 3 (Perl code reading): 10 marks

Consider the following code.

```
#!/usr/bin/perl -w

use strict;

my %login2sid;

while (my $line = <STDIN>) {
     chomp $line;
     my ($login, $sid) = split q{:}, $line;
     if ($sid < 80000) {
          $login = "oldalumni_" . $login;
     }
     $login2sid{$login} = $sid;
}

my @logins = keys %login2sid;
foreach my $s (sort @logins) {
     print "sid ", $login2sid{$s}, " --> ", $s, "\n";
}
```

And here is some sample input piped into stdin:

```
stephaned:812133
sharper:76011
gwbush:0183220
spsqpants:9963219
mmouse:61301
britneys:9089119
```

What is the output? Write each line in the space provided below.

Question 4 (Perl code writing): 20 marks

The text file shown below uses the following format:

```
<input file>:<arguments>:<expected output>
```

For example, here is such a file:

```
napoleon.txt:-w 20 -i 1:out200.txt
napoleon.txt:-w 25 -i 2:out201.txt
napoleon.txt:-w 30 -i 5:why314.txt
meeting.txt:-i 10 -n -w 45 -p:good202.txt
```

Write a Perl script using such text files to drive the testing of a script named `your_indent.pl`:

- “`<input>`” is redirected into “`your_indent.pl`”
- “`<arguments>`” are passed to the script
- “`<expected output>`” is the name of the file containing the correct result
- the only command-line argument provided to the script is the directory containing the `<expected output>` files

Your script’s output should be the number of tests that pass, i.e., where the expected output is exactly matched by the output from `your_indent.pl`. Some marks will be given for the quality of your solution. Use the `system()` function in Perl to invoke shell commands (i.e., a string passed to `system()` is interpreted as a Unix command; for example, `system(“cat foo.txt | diff bar.txt -”)` will compare the contents of foo.txt with bar.txt, and if a different is detected, a non-zero value will be returned from `system()`.)

Question 5 (C code reading): 24 marks

a) What is the output of the following program *as it is written*? Provide your answer in the line provided below.

```c
#include <stdio.h>
int main (void)
{
        int a = 1;
        int b = 2;
        int c = 3;

        int d = a * b;

        if (d = c) {
                a = a + b;
        } else {
                a = 30;
        }
        printf(“%d, %d, %d, %d\n”, a, b, c, d);
}
```

Output:

____

Question 5 (C code reading) continued

b) What is the output of the following program? Write your answer in the lines provided below.

```c
#include <stdio.h>

int c = 99;

int mystery(int a, int b)
{
        int temp;
        temp = a;
        a = b;
        b = temp;

        return temp;
}

int main()
{
        int x = 10;
        int y = 20;
        int z = 30;
        printf("%d %d %d %d\n", c, x, y, z);
        c = mystery(x, y);
        printf("%d %d %d %d\n", c, x, y, z);
        c = mystery(z, c);
        printf("%d %d %d %d\n", c, x, y, z);
}
```

Output

Question 5 (C code reading) continued

c) What is the output of the following program? Write your answer in the lines provided below.

```
#include <stdio.h>

int main()
{
        int a[4] = {5, 10, 15, 11};
        int b = 1050;
        int *p1 = &b;
        int *p2 = a;
        int *p3 = &a[2];

        printf("%d %d %d %d %d %d %d\n", a[0], a[1], a[2],
            b, *p1, *p2, *p3);

    *p2 = 44;
    p3 = p1;
    *p3 = 55;
    printf("%d %d %d %d %d %d %d\n", a[0], a[1], a[2],
            b, *p1, *p2, *p3);

    a[2] = 99;
    *p3 = 219;
    printf("%d %d %d %d %d %d %d\n", a[0], a[1], a[2],
            b, *p1, *p2, *p3);

    *(p2 + 1) = 555;
    a[0] = 665;
    printf("%d %d %d %d %d %d %d\n", a[0], a[1], a[2],
            b, *p1, *p2, *p3);
}
```

Output

Question 6 (Software Engineering methodology): 18 marks

One of the Principles of Software Development is that you must "expect to deal with change". Describe three things you have learned from this course that can be used to observe this principle. For each, give a brief example justifying your answer. Use diagrams, point form and code where appropriate. Clearly label each of the three parts of your answer.

Question 6 continued

Question 7 (C types and `malloc`): 30 marks

a) Construct (an) ANSI C aggregate type(s) to be used for storing music MP3 playlist information. You can assume there are no more than 99 songs in a playlist; each song will have title info, a file location, plus a playing length in minutes and seconds; each playlist has a title and running length; and each playlist also has an a genre. Some marks will be given for the quality of your solution.

b) Please declare and define the function `init_playlist(int n)` which dynamically allocates memory for a playlist where the number of songs in the playlist equal to `n`. A pointer is passed back as a return value.

c) Consider a playlist stored in a file named `265.list` where this file consists of the following lines of text:

```
title#Greatest Lectures of SENG 265
genre#spoken word
length#3
Why Perl is glorious!#/Volume/MP3/GLS265/lecture12.mpg#50#12
Development processes: Why?#/Volume/MP3/GLS265/lecture9.mpg#45#00
Coding at Voice Mobility#/Volume/MP3/Guests/paul.mpg#49#20
```

Write code which would add this information to a playlist and compute the running length. You can assume the file above is redirected into `stdin` of your program. Some marks will be given for the quality of your solution.

Question 8 (Debugging): 10 marks

a) Give two small examples, along with explanations, of *observable failures*.

b) The Knuth/Barr bug categories are "A" for *Algorithm*, "D" for *Data*, "F" for *Forgotten*, and "B" for *Blunder*. What is a major difference between A-category bugs and F-category bugs?

END OF EXAM