

Given name:\_\_\_\_\_ Family name:\_\_\_\_\_

Student number:\_\_\_\_\_ Signature:\_\_\_\_\_

**UNIVERSITY OF VICTORIA**

**Faculty of Engineering**

**Department of Computer Science**

**Examination December 2001**

**SEng 265 (Introduction to Software Engineering)**

**Instructor: Daniel M. German**

**Duration: 3 hrs**

**No books or electronic devices are allowed**

**This exam is to be answered on the examination paper itself**

This examination paper consists of **13** pages and **7** questions. Please bring any discrepancy to the attention of an invigilator. The number in brackets at the start of each question is the number of points the question is worth.

Answer all questions.

**Please write your answers clearly.**

For instructor's use:

	Score
1 (15)	
2 (16)	
3 (10)	
4 (9)	
5 (10)	
6 (10)	
7 (10)	
Total (80)	

## 1. C Questions

This section is worth 14 points.

Assume you have the following function definition:

```
#include <stdio.h>

char *pr (char *str)
{
    char *pc;
    pc = str;
    while (*pc)
        putchar(*pc++);

    do {
        putchar(*--pc);
    } while (pc - str);
    putchar('\n');
    return (pc);
}
```

Consider the following function call:

```
x = pr("UVic!");
```

(a) [3] What is printed?

(b) [2] What type should x be?

(c) [3] What value does `x` get?

(d) [4] What does the expression `*--pc` mean, and how is it different from `--*pc`?

(e) [3] What would be printed if `*--pc` were replaced with `*pc--`?

## 2. Expressions

This section is worth 16 points.

- (a) [8] C Language. What are the values of *i* and *j* after each of the following sequences of statements? Sequences of statements are not cumulative.

```
int i=0, j=0;
i = 3 * 2+4; j = 28 % 3;
```

*i*: \_\_\_\_\_ *j*: \_\_\_\_\_

```
int i=-1, j=0;
int *p=&i;
++i;
++*p;
j = *p ? 1 : 2 ;
```

*i*: \_\_\_\_\_ *j*: \_\_\_\_\_

```
int i,j;
i = 3;
j = --i;
(i--, j++);
```

*i*: \_\_\_\_\_ *j*: \_\_\_\_\_

```
int i,j;
i = (j = 5) + 3;
j*=2;
```

*i*: \_\_\_\_\_ *j*: \_\_\_\_\_

- (b) [8] Perl Expressions. What are the values of `$i` and `$j` after each of the following sequences of statements? Sequences of statements are not cumulative.

```
$i=0, $j=0;
$i = "This is the value of $i";
$j = 'This is the value of $j';
$i =~ /This is/;
$j =~ s/This is/not/;
```

`$i:` \_\_\_\_\_ `$j:` \_\_\_\_\_

```
($i, $j) = ("Abc" . "d", "efg");
```

`$i:` \_\_\_\_\_ `$j:` \_\_\_\_\_

```
$i = 3;
$j = --$i;
($i--, $j++);
```

`$i:` \_\_\_\_\_ `$j:` \_\_\_\_\_

```
$i=-1;
$j=0;
$k = -1;
$j = ++$k? ++$i : $i++ ;
```

`$i:` \_\_\_\_\_ `$j:` \_\_\_\_\_

### 3. Writing C Code

- (a) [10] Using the following prototype, write a function that computes the average, maximum and minimum of a set of numbers.

```

/*
 * Name:      Average_Max_Min
 *
 * Purpose:   To compute the average, maximum and minimum of a set
 *            of integers
 * Parameters:
 *   Input:
 *       inputSet : an array of integers
 *       n        : the number of integers in the array
 *       pMax     : a valid pointer to an integer
 *       pMin     : a valid pointer to an integer
 *   Output:
 *       pMax     : the integer it points to contains the maximum
 *                  value in the array
 *       pMin     : the integer it points to contains the minimum
 *                  value in the array
 *
 * Returns:
 *       the average of the set
 *
 * Comments:
 *       This function should be called with n > 0
 */
int Average_Max_Min(int inputSet[], int n, int *pMax, int *pMin);

```

[This page left blank for your answers]

#### 4. Perl

This section is worth 9 points

(a) [3] What would the following code print?

```
@array = (5, 4, 3, 1, 2);  
foreach (@array) {  
    print;  
}
```

(b) [3] What would the following code print?

```
$x = "1";  
$y = 'a';  
print 'echo "Hello world"' if (1 == $x);  
print 'echo "Hello world"' if ("a" == $y);  
print "echo 'Hello world'" if ("a" eq $y);
```



## 5. Perl Regular Expressions

This section is worth 10 points. For this section assume we have the following 5 strings:

```
Visual Age for C++  
Visual Age for Java  
^GNU/Linux  
GNU Emacs Pocket Reference  
gcc++
```

For each of the following regular expressions, list the strings that would be matched by it.

(a) [2] /g.\*\$/i

(b) [2] /c\+ /

(c) [2] /c+ /

(d) [2] /^GNU\ /Linux\$/

(e) [2] /[<sup>^</sup>G]+NU/

## 6. Testing and Debugging

This section is worth 10 points.

The following section of code is expected to print all the characters of a string, one per line.

```
i = 0;
do {
    putchar(s[i++]);
    putchar('\n');
} while (s[i] != '\0');
```

- (a) [5] Doing boundary testing, find a test case (a value of *s*) under which the section of code does not work.

- (b) [5] Fix the section of code, so it does what it is supposed to do (prints all the characters of a string, one per line).

## 7. Make

This section is worth 10 points. For this section, use the following Makefile and the following files:

```
default: c

a.o: a.c a.h b.h
    gcc -o a.o -c a.c

c.o: c.c a.h
    gcc -o c.o -c c.c

b.o: b.c b.h
    gcc -o b.o -c b.c

c: a.o c.o b.o
    gcc -o c a.o c.o b.o

clean:
    rm -f *.o c *~
```

Name	Time of last modification
a.c	Dec 2 23:03
a.h	Dec 2 23:05
a.o	Dec 2 23:22
b.c	Dec 2 23:03
b.h	Dec 2 23:04
b.o	Dec 2 23:22
c	Dec 2 23:22
c.c	Dec 2 23:11
c.o	Dec 2 23:22
Makefile	Dec 2 23:11

The following questions are not cumulative (i.e. in order to answer a given question, you should not take into account your answers to the previous questions).

- (a) [2] You modify the file a.h, then you type `make`. Which commands will be executed and in which order?

- (b) [2] You modify the file `a.c`, then you type `make`. Which commands will be executed and in which order?
- (c) [2] You modify the file `b.c`, then you type `make`. Which commands will be executed and in which order?
- (d) [2] You modify the file `b.h`, then you type `make`. Which commands will be executed and in which order?
- (e) [2] You modify the file `c.c`, then you type `make`. Which commands will be executed and in which order?

## End of examination

**Total pages: 13**

**Total marks: 80**