VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

EXAMINATIONS — 2006

END-YEAR

COMP 206

PROGRAM AND DATA STRUCTURES

Time Allowed: 3 Hours (180 minutes)

Instructions:
- Attempt all questions.
- There are 180 possible marks on the exam.
- Make sure your answers are clear and to the point.
- Non-programmable calculators without full alphabetic keys are permitted.
- Non-electronic foreign language dictionaries are permitted.
- Refer to the Appendix.
- No other reference material is allowed.
- Answer in the appropriate outlined boxes or follow the instructions given in the questions.

Question	Topic	Marks
PART 1		
1	C Basics	[16 marks]
2	Dynamic Data Structures	[20 marks]
PART 2		
3	Comparing C and C++	[18 marks]
4	C++ Programming	[30 marks]
5	C++ Language	[16 marks]
PART 3		
6	File Structure Fundamentals	[25 marks]
7	B-tree	[20 marks]
8	Index-Sequential File	[15 marks]
9	Secondary Indices	[10 marks]
10	Hash File	[10 marks]

StudentId_____

PART 1: C

Question 1. C Basics [16 marks]

a) [1 mark] In the box below give a suitable declaration in C of a pointer to an integer:

b) [1 mark] In the box below give a suitable declaration in C of a function which takes a pointer to a character and returns a pointer to an integer:

c) [1 mark] In the box below give a suitable declaration in C of an 8 element array of pointers to integers:

d) [1 mark] In the box below give a suitable declaration in C of a pointer to an 8 element integer array:

e) [3 marks] I compiled the following C code using gcc 4.0.0 on a Macintosh with a PowerPC G5 chip:

```
#include <stdio.h>


int main()
{
   printf("Size of int* = %d\n", sizeof(int*));
   return 0;
}
```

It produces the following output:

```
Size of int* = 4
```

Suppose I change `int` to `double`, to get this program:

```
#include <stdio.h>


int main()
{
   printf("Size of double* = %d\n", sizeof(double*));
   return 0;
}
```

In the box below, state the output of the new program, and explain why this answer is produced.

ANSWER

f) [3 marks] In the box below, state the output that will be generated by the following C program .

```
#include <stdio.h>

int f(char* s);

int main()
{
    char* s;

    s = "COMP206";

    printf("f(%s) = %d\n", s, f(s));
    return 0;
}

int f(char* s){
    char *p = s;

    while(*p!='\0')
        p++;

    return p-s;
}
```

ANSWER

g) [6 marks] Briefly explain the differences between static and automatic storage classes in C.

ANSWER

Question 2. Dynamic Data Structures [20 marks]

A linked list is a dynamic data structure. Each item in a linked list has some data and has a pointer to the next item in the list.

You want to keep track of your book collection. You decide that the important information about a book is:

- the title,
- the author, and
- price.

a) [5 marks] Give a suitable C declaration (including comments) for the type of a linked list of books.

ANSWER

b) [5 marks] Write C code (including comments) to implement a function `count` to count the number of books in a linked list of books.

ANSWER

c) [5 marks] Write C code (including comments) to implement a function `addbook` which takes a title, an author and a price and adds a new book to a linked list of books. The new item should be the first item in the list.

ANSWER

d) [5 marks] Write C code (including comments) to implement a function `removebk` to remove the first book from a linked list of books.

ANSWER

PART 2: C++

Question 3. Comparing C and C++ [18 marks]

a) [6 marks] Describe the ways in which parameters can be passed to functions in C and in C++. Be careful to highlight the differences between the two languages.

ANSWER

b) [12 marks] C provides `malloc` and `free`. C++ provides `malloc`, `free`, `new` and `delete`.

Explain what `malloc`, `free`, `new` and `delete` do and explain why C++ provides all of them.

ANSWER

Question 4. C++ Programming [30 marks]

You are working as a programmer for a company which operates a fleet of taxis. The company wants to run a simulation in order better to understand how much use their taxis are getting. In their simulation they want to know:

- for each taxi, how long its current trip is;
- for each taxi, how far it has travelled in all trips;
- how far all the taxis combined have travelled.

You decide to write a C++ class called `Taxi`. The taxi class must:

- record the appropriate data
- support:
  - creation of instances;
  - assignment;
  - incrementing;
  - output;
  - input of a trip distance;
  - resetting of the trip.

For example, given the following `main` function:

```
int main(){
  Taxi t1;           //create t1

  Taxi t2 = t1;

  cin >> t1;          //read some trip data from the terminal
  t1++;               //t1 travels 1km
  t2++;               //t2 travels 1km
  cout << t1 << endl; //t1 reports its state
  t1.reset();         //t1's trip is over
  cout << t1 << endl << t2 << endl;
}
```

the compiled code should have this behaviour (user input is underlined):

```
How far? 5
Trip: 6 My total: 6 Combined total: 7
Trip: 0 My total: 6 Combined total: 7
Trip: 1 My total: 1 Combined total: 7
```

a) [10 marks] Complete the declaration in the box below to define a suitable C++ class `Taxi`. You do not need to implement the methods.

ANSWER

```
class Taxi {




};
```

b) [5 marks] In the box below give C++ code (including comments) for the assignment operator for `Taxi`:

ANSWER

c) [5 marks] In the box below give C++ code (including comments) for the post-increment operator for `Taxi`:

ANSWER

d) [5 marks] In the box give C++ code (including comments) for the << operator for `Taxi`:

ANSWER

e) [5 marks] In the box below give C++ code (including comments) for the >> operator for `Taxi`:

ANSWER

Question 5. C++ Language [16 marks]

a) [8 marks] C++ provides virtual functions. In the box below explain what benefits virtual functions provide for the programmer.

ANSWER

b) [8 marks] C++ provides templates. In the box below explain what benefits templates provide for the programmer.

ANSWER

SPARE PAGE FOR EXTRA ANSWERS

Cross out the rough working that you do not want marked.
Specify the question number for work you do want marked.

PART 3: File Structures

Question 6. File Structure Fundamentals [25 marks]

a) [2 marks] Define the term “file record format”.

ANSWER

b) [2 marks] Define the term “record key”**.**

ANSWER

c) [9 marks] Describe each of the three basic file organizations using the following two criteria:

- The way file records are assigned to storage locations, and
- The relationship between a record’s key value and the relative address of the location the record is stored in.

ANSWER

d) [12 marks] Suppose the declarations of a `struct` and a variable given below are defined in a program.

```
typedef struct {
     int StudentId;
     char Name[16];
     char Address[31];
     } Record;
     …
     Record student;//student variable
```

Suppose there is the following command

```
FILE *sptr = fopen("student.data", "w");
```

and it returns a non-`NULL` `sptr` value.

I. [3 marks] Suppose the `student.data` file contains records of a predictable length. Use the `fprintf` C Stream function to write a student record into the `student.data` file.

ANSWER

II. [9 marks] Suppose the `student.data` file contains records with length indicators in front of each record and each field. Write a part of a C program that will compute the actual length of each field and the record itself and then use the `fprintf` C Stream function to write a student record into the `student.data` file.

ANSWER

StudentId_____

SPARE PAGE FOR EXTRA ANSWERS

Cross out the rough working that you do not want marked.
Specify the question number for work you do want marked.

Question 7. B-tree [20 marks]

a) [8 marks] In a B-tree of order $p = 2m + 1$ and height h:

I. [2 marks] What is the minimum number of (key, address) pairs in a node that is not the root?

ANSWER

II. [2 marks] What is the maximum number of (key, address) pairs in a node?

ANSWER

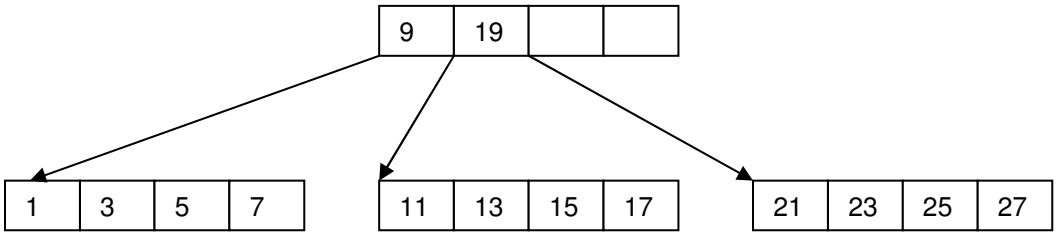
III. [2 marks] What is the minimum number of (key, address) pairs in the root node?

ANSWER

IV. [2 marks] What is the number of edges between the root and a leaf node expressed in terms of the height h?

ANSWER

b) [7 marks] Consider the B-tree of order 5 below. The address components of node entries are omitted for the sake of simplicity.



In the box below, show the B-tree after inserting the key value 16.

ANSWER

c) [5 marks] The Admin Node of a B-tree file similar to one you have seen in Assignment 3 contains the following fields:

- `int num_records // number of records in the file,`
- `int num_of_nodes // number of actual tree nodes,`
- `int num_of_blocks // number of blocks allocated so far to the file,`
- `int ROOT // the relative address of the root node.`

The variable `node_size` contains the length of a node. The file is implemented as a binary file.

Suppose a node splits. How does the `btree.cpp` program compute the relative address of the new node?

ANSWER

Question 8. Index-Sequential File [15 marks]

The file header of an index-sequential file with a B-tree is stored in a file allocation table in main memory. The file header contains various information about the file, such as: number of blocks allocated to the file, the address of the B-tree root node, the address of the left most sequence set, and the number of records in the file. The file contains $r = 65000$ records. File records have a fixed size of $L = 300$ bytes. File blocks have a size of $B = 4096$ bytes. Each block has a header of $d = 96$ *bytes*.

a) [3 marks] Calculate the range of values of the number s of sequence sets.

ANSWER

b) [12 marks] The file is processed sequentially. The average access time to a sequence set (contained in a block on disc) is *3 ms*, the time to read a block into the main memory buffer is *2 ms*, and the time to process a sequence set is *4 ms*.

I. [3 marks] Suppose there is only one buffer of *4096* bytes allocated to the index-sequential file. Calculate the expected time to process the file in the worst case.

ANSWER

II. [9 marks] Suppose there are two buffers of *4096* bytes allocated to the index-sequential file. Calculate the expected time to process the file in the worst case.

ANSWER

Question 9. Secondary Indices [10 marks]

Suppose:

- An *Exam* file contains *r = 90000* records,
- Record format is *Exam*(*int StudentId, char CourseId, char Term, char Grade*),
- The file is stored on disk and its structure consists of a data area and several indices,
- There is a single level secondary index on *Grade* containing *10* secondary key entries,
- There is a secondary single level index on *CourseId* having *200* secondary key entries,
- There is a single level secondary index on *Term* having *10* secondary key entries,
- All pointers are *p = 8* bytes long,
- The file block size is *B = 8192* bytes, and each block contains a pointer to the next level of indirection,
- All distributions are even, and
- There are several records containing *Grade* = "*A+*", or *CourseId* = "*COMP206*", or *Term* = "*2006T2*" in the file.

How many accesses to disk will it be needed to evaluate the query

Retrieve all exam records having Grade ="A+" AND CourseId = "COMP206" AND Term= "2006T2".

ANSWER

Question 10. Hash File [10 marks]

a) [2 marks] What is a hash function?

ANSWER

b) [2 marks] What are synonyms?

ANSWER

c) [2 marks] What is a bucket?

ANSWER

d) [2 marks] What is the home bucket of a record?

ANSWER

e) [2 marks] What is an overflow record?

ANSWER

****

APPENDIX

Low Level I/O System Calls:

```
int fd = open(const char file_name,
                                int flags, [mode_t pmode]);
```

- `flags (O_RDWR | O_RDONLY | O_WRONLY, [O_CREATE], [O_APPEND], [O_TRUNC],...)`

```
ssize_t retval = write(fd, source, size);

ssize_t retval = read(fd, dest, size);

off_t seekval = lseek(int fd,
                            off_t offset, int reference);
```

- `reference – SEEK_SET | SEEK_CUR | SEEK_END`

C Stream File I/O Commands (Text File)

```
FILE *sptr = fopen(char file_name, char file_type);
int fprintf(FILE *sptr, control_string, arg1,…, argn);
int fscanf(sptr, control_string, arg1,…, argn)
```

- `– control string – formatting information,`
- *argi* (*1 ≤ i ≤ n*) - individual output data items

```
long seekval = fseek(FILE *spr, long offset, int ref);
```

- `ref – 0 for SEEK_SET, 1 for SEEK_CUR, 2 for SEEK_END`

File Performance Formulae:

blocking factor $f = \lfloor (B - header)/L \rfloor$
number of blocks $b = \lceil r/f \rceil$
external sort-merge $N = 2b(1 + \lceil (\log_{n-1} b) - 1 \rceil)$
number of buffers n

B-tree (the worst case)

$h = 1 + \lfloor \log_{m+1}((r+1)/2) \rfloor$
number of leaves = $2(m+1)^{h-2}$

B$^+$-tree (the worst case)

$h = 2 + \lfloor \log_{m+1}(r/2m) \rfloor$
number of leaves = r/m

Index-Sequential File with a B-tree

number of sequence sets s
$\lceil r/f \rceil \le s \le \lceil 2r/f \rceil$