CPSC 200 Fall 2001
Midterm II—05 November 2001

Name(Printed) :	
Signature :	
StudentNumber :	

- Write the word circled above on each page of your exam. Do not put any other identifying marks on any page of your exam. Failure to put the circled word on a page of your exam may result in no marks being awarded for that page.
- *Read each question carefully. Ask yourself what the point of the question is. Check to make sure that you have answered the question asked.*
- This is a **50** minute exam. This exam contains **7** pages of questions not including this cover page. Make sure that you have all of them.
- Answer all questions on the exam sheet. If you do some of your work on the back of a page, clearly indicate to the marker what work corresponds with which question.
- Partial marks shall be awarded for clearly identified work.
- This exam counts as **20%** of your total grade. There are **50** points total on the exam.

ACRE	AREA	BALE	BAND	BARD	BASS
BETA	BIRD	BLOT	BOOK	BREW	CAMP
CHIP	CLAN	COAT	COIL	CORN	CROW
CURL	DARK	DEER	DOSE	DROP	DUCK
DUSK	FARE	FILM	FLAX	GAZE	GIFT
GOLD	GULF	HINT	HORN	HULL	IBOU
INCH	IRIS	ISLE	KERN	KILN	KITE
LANE	LARK	LENS	LOFT	LURE	MALT
MANX	MESH	MINK	MOTH	MOVE	MUSK
NAVY	NEWT	NOON	OATS	OBOE	OPAL
PARK	PINE	POET	RAFT	REED	RING
RUBY	RUFF	SEAM	SEED	SHOP	SILK
SINE	SNIP	SOAP	STUB	TASK	TAXI
TEAM	TELL	TEXT	TIDE	TILT	TOIL
TOME	TOUR	TURN	VANE	VISA	WALL
WICK	WOLF	WRIT	YARD		

Sums

$$
\begin{aligned}
\sum_{i=1}^{n} i &= n(n+1)/2,\\
\sum_{i=1}^{n} i^2 &= (n+1)^3/3-(n+1)^2/2+(n+1)/6,\\
\sum_{i=1}^{n} i^3 &= (n+1)^4/4-(n+1)^3/2+(n+1)^2/4=\left(\sum_{i=1}^{n} i\right)^2,\\
\sum_{i=1}^{n} 2^i &= 2^{n+1}-1.
\end{aligned}
$$

Logarithms and Exponential

$$a^x = y \qquad \Leftrightarrow \qquad x = \log_a y$$

$$\log a^s b^t = s\log a + t\log b, \qquad (x^a)^s\cdot\left(x^b\right)^t = x^{as+bt}$$

in particular $\log ab = \log a + \log b$, $\log a/b = \log a - \log b$, $\log a^s = s\log a$. To change from logarithms base a to logarithms base b use

$$\log_b x = \frac{\log_a x}{\log_a b}.$$

Factorials

$$n! = n\cdot(n-1)! = \prod_{i=1}^{n} i \qquad 5! = 5\cdot4\cdot3\cdot2\cdot1 = 120.$$

Sorting

(5) **1.** Fill in the following table:

Algorithm	Average-case Θ-time	Worst-case Θ-time	Extra storage requirements	stable (yes / no)
insertion sort				
merge sort				
quick sort				
Shell sort Sedgewick's increments				

2. Suppose that you are trying to sort an array of 4 doubles in a minimum number of comparison.

(2) Can you always sort 4 elements with at most 5 comparisons? Why or why not?

(2) **(b)** Give one method for sorting an array of 4 elements that uses the least possible number of comparisons for the worst case.

3. The parts of the following question refer to the list:

$[9, 8, 10, 7, 18, 8, 11, 9, 10, 8]$.

(2) (a) How many inversions are there in the list?

(2) (b) Partition the list (as in quick sort) using **9** as your pivot value.

(2) (c) 3-sort the list (as in Shell sort).

(2) (d) Stably sort the list using the predicate

```
bool lt(int a, int b) { return (a%2)<(b%2) ; }.
```

4. Give defintions for the following concepts:

(2) (a) An inversion.

(2) (b) A stable sort.

(2) (c) A sentinel element.

5. Give formulæ for:

(1) (a) the average number of inversions in an array of size n.

(1) (b) the best-case number of inversions in an array of size n.

(1) (c) the running time (as a Θ-function of the array size, n) to form a heap.

(1) (d) the running time to k-sort an unsorted array of size n.

(1) (e) the running time of insertion sort as a Θ-function of the number of inversions I and the size of the array n.

(2) **6.** Why do Shell's increments for Shell sort work so poorly?

(2) **7.** In order to save on storage costs and more correctly reflect the accuracy of its thermometers a weather station switches from storing temperature values (in degrees Celsius) in `double`s to storing them in `short int`s. Surprisingly, this slows down data processing, and it is discovered that a home grown quick-sort algorithm is primarily responsible for the increased running time. What is likely wrong with the quick-sort algorithm?

Templates and the Standard Library

(5) **8.** Name five major groupings of the components of the Standard Template Library.

9. The code shown in Figure 1 has been modified and rearranged from the STLPort implementation of the STL.

(1) (a) `reverse` is an example of which of the 5 main categories of STL components?

Adaptation of STLPort's `reverse` code

```
template <class Iterator>
void reverse(Iterator first, Iterator last)
{
  for (;first != last && first != --last;++first)
      {
      swap(*first, *last);
      }
  return ;
}
```

Figure 1: Code for question 9

(2) (b) `reverse` takes two iterators as arguments, rather than one argument of a type something like a `vector`. How is the choice of arguments consistent with the STL design philosophy? Why is this approach better or worse than passing something like a `vector` directly to `reverse`?

(1) (c) What kind of iterators do `first` and `last` need to be in order for this algorithm to function?

(3) (d) What are 3 other standard classes of iterators?

(2) (e) Note that `last` is decremented before performing the first swap, whereas `first` is not. This is consistent with the conventional STL way of pass-

ing pairs of iterators to functions. What convention does the STL follow when passign pairs of iterators to a function?

(1) (f) The loop test is written as "`first != last && first != --last`". Why not just write "`first < --last`"?

Error handling

(2) **10.** (a) What makes error handling such a difficult topic?

(2) (b) When is a "`throw;`" statement legal?

(1) (c) What restrictions are there on the order of `catch`-blocks after a `try`-block?

(3) (d) Explain when and how destructors are executed as a consequence of executing `try-throw-catch` logic.

Question	Score
1	/5
2	/4
3	/8
4	/6
5	/5
6	/2
7	/2
8	/5
9	/10
10	/8
Total	/55