Computer Science 1 Bh

Degree Examination

Date: Saturday 27th May 2000
Time: 12:00–13:30 (one and a half hours)
Place: Roxburgh Place
Room: Halls A and B

Board of examiners
Chair: Stuart Anderson.
External Examiner: Muffy Calder.

Instructions to candidates
1. Check that the question paper contains pages 1–11 after this cover page. If it does not, inform the invigilator.
2. You should attempt as many questions as possible.
3. The questions in this paper vary widely in difficulty. During your first pass through the paper, you are advised not to dwell on questions to which the answer is not readily apparent.
4. Write your answer to each question in the box or table provided. If you wish to write more in answer to a question, continue on the blank page opposite, indicating that you have done so.
5. Please write legible and concise answers.
6. The marks allocated to each part of a question are indicated in the margin. There are 100 marks in total.
**Question 1**

The Java language provides a method `Math.pow()` for exponentiation. If \( x \) is a floating-point number and \( n \) is an integer then

\[
\text{Math.pow}(x, n) = x^n
\]

(a) Suppose that the `Math.pow()` method is not available. Using only the operations of addition, subtraction and multiplication, by repeated multiplication or otherwise, write a Java implementation of a method

\[
\text{float power(float x, int n)}
\]

such that, if \( x \) is a floating-point number and \( n \) is a positive integer, then \( \text{power}(x, n) = x^n \).

[4 marks]

(b) Use induction to show that your function is correct. State clearly the induction hypothesis, base case, and induction step.

[6 marks]
Question 2

This question concerns stack-based evaluation of arithmetic expressions.

(a) Draw syntax trees for the following expressions:
   • 7 × (5 + 3)
   • (7 × 5) + 3
   • (5 + 3) × (6 + 4)  [3 marks]

(b) Give the expressions derived from each of these trees by (a) infix and (b) postfix tree traversal.  [3 marks]

(c) Describe the instructions and operation of a simple abstract stack-machine suitable for evaluating such expressions. Using the expression (5 + 3) × (6 + 4) as an example explain how to generate instructions for your machine.  [4 marks]
Question 3

(a) Explain what it means for a valuation, \( V \), to satisfy a Boolean propositional formula, \( \varphi \).

\[ 2 \text{ marks} \]

(b) The remainder of this question concerns the Davis-Putnam algorithm for finding a satisfying valuation for a formula, \( \varphi \).

i. Describe the two reductions used by the algorithm.

\[ 2 \text{ marks} \]

ii. Explain carefully why these reductions are sound.

(You should state the relevant relationship between the formula and its reduction, but are not asked to prove that it holds.)

\[ 2 \text{ marks} \]

This question is continued on the following page
(c) Apply the Davis-Putnam reductions to the following formulae, showing each reduced formula and the corresponding partial valuation on a separate line.

i. \((A ∨ B ∨ D) ∧ (A ∨ C) ∧ (¬B) ∧ (¬D ∨ ¬C)\)  

ii. \((A ∨ B ∨ D) ∧ (¬A ∨ C) ∧ (¬B) ∧ (¬D ∨ C) ∧ (B ∨ ¬A)\)
Question 4

(a) In addition to applications, the Java programming language provides the programmer with the ability to define restricted programs called applets. List three things which a Java application could do but a Java applet could not. [3 marks]

(b) Why are the sandboxing restrictions on applets considered to be a good idea? [3 marks]

(c) Give an example of a problem which a hostile applet could cause, despite sandboxing. [3 marks]

(d) The following method is special for Java applications because it is never normally invoked by the programmer.

```java
public static void main (String[] args)
```

Give the heading of a method which is similarly special for Java applets. As above, include the access control modifiers, the return type and the formal parameter list. [1 mark]
Question 5

(a) Supply a Java method which would produce the following Java byte code method when compiled.

```
Method int subtract(int, int)
  0  goto 9
  3  iinc 1 -1
  6  iinc 2 -1
  9  iload_1
 10  if_ne 3
 11  iload_2
 12  ireturn
```

(b) Supply the compiled Java byte code equivalent of the following Java method.

```
int abs (int x) {
  if (x < 0)
    return -x;
  else
    return x;
}
```
Question 6

(a) A Java program contains the following statement where `ArrayIndexOutOfBoundsException` is the exception thrown for illegal array access and `StringIndexOutOfBoundsException` is an exception which can be thrown by the `charAt()` method.

```java
try {
    System.out.println(args[2].charAt(2));
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("array error");
} catch (StringIndexOutOfBoundsException e) {
    System.out.println("index error");
}
```

Supply the character or string which is printed when the array `args` has the values shown below.

<table>
<thead>
<tr>
<th>Contents of the array <code>args</code></th>
<th>String or character printed</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ &quot;&quot; }</td>
<td></td>
</tr>
<tr>
<td>{ &quot;ab&quot;, &quot;cd&quot; }</td>
<td></td>
</tr>
<tr>
<td>{ &quot;abc&quot;, &quot;def&quot;, &quot;ghi&quot; }</td>
<td></td>
</tr>
<tr>
<td>{ &quot;&quot;, &quot;&quot;, &quot;abcdef&quot; }</td>
<td></td>
</tr>
</tbody>
</table>

(b) Rewrite the `try-catch` statement above using `if-else` statements. Your code should print the strings `array error` and `index error` if the statement above would do so and the appropriate character from the string otherwise.
Question 7

The skeleton method given below is intended to determine the smallest number of coins whose values are given in the vector coin to change the amount \( n \). The method is intended to use recursion to calculate the answer. Your task is to fill out the boxes in the skeleton with code which defines the method correctly.

```java
public int change(Vector coin, int i, int n) {
    // Returns the number of coins needed to change amount n
    // using coins from i of coin upwards.
    if (n == 0) {
        // We have changed the amount exactly.
    }
    else if (i == coin.size()) {
        // We have run out of coins
    }
    else {
        // Compute the value of the next coin.
        int coinI = ((Integer)coin.elementAt(i)).intValue();
        if (coinI > n) {
            // The next coin is too big to use in changing n.
        }
        else {
            // We can choose either to use the ith coin or not.
        }
    }
}
```

[2 marks]
[2 marks]
[2 marks]
[4 marks]
Question 8

(a) The method given below is intended to find the minimum of a non-empty vector of Integer objects.

```java
public int min(Vector v) {
    int result = 0;
    for (int i = 0 ; i < v.size() ; i++) {
        int val = ((Integer)v.elementAt(i)).intValue();
        if (val < result) {
            result = val;
        }
    }
    return result;
}
```

i. Describe a sequence of integers that will give the correct result using the above program. [2 marks]

ii. Describe a sequence of integers that will give incorrect results using the above program. [2 marks]

iii. Outline how you would go about repairing the fault in the program so it would give correct results with both your tests. [2 marks]

(b) Briefly describe three different reasons for testing programs. [4 marks]
Question 9

(a) How many different ways or permutations are there of scheduling four tasks on three processors?

(b) You are given eight tasks which take the following amounts of time to run.

<table>
<thead>
<tr>
<th>Task number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>7</td>
<td>20</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

They are to be scheduled on three processors so that each processor gets roughly an equal amount of work to do.

i. How would the round robin scheduling technique schedule these tasks on three processors?

ii. How would the greedy algorithm scheduling technique schedule them on three processors?
Question 10

(a) Why is parallelism useful? [2 marks]

(b) Explain what is meant by data parallelism. [2 marks]

(c) Give one example of pipeline parallelism. [1 mark]

(d) What is the sequential time complexity of quicksort? [2 marks]

(e) If we apply a parallel divide-and-conquer approach to quicksort and have \( n \) processors, what is the parallel time complexity? [3 marks]