Closed Book Examination

Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Imperative Programming with C and C++

Friday 30\textsuperscript{th} January 2009    Time: 14:00 – 16:00

Please answer any THREE questions from the FOUR questions provided

Use a SEPARATE answerbook for EACH question

The use of electronic calculators is NOT permitted
1. a) Describe the structure and syntax of a function declaration in C, paying particular attention to the definition of parameters and return value types. (3 marks)

b) What is the role of a function prototype, and what is the meaning of the term *formal parameter*? Give an example of a function prototype declaration. (3 marks)

c) Is there anything wrong with the following code? (3 marks)

```c
#include <math.h>
void main()
{
    int b;
    float a;
    b = nearest_int(a);
}
int nearest_int(float a)
{
    return ((int) round(a));
}
```

d) The following C function may work correctly and return a copy of the string passed as a parameter. However, there are four potentially serious flaws: what are they? (6 marks)

```c
char *strcopy(char *string)
{
    int i = 0;
    char copy[20];
    while (string[i] != '\0') copy[i] = string[i++];
    return copy;
}
```

e) Write a function which alters the value of the parameter passed to it, by replacing it by the square of its value. Include in your answer code to show how the function should be called. (5 marks)
2.  
   a) Discuss the relationship between pointers and arrays in C. As part of your answer, explain what is meant by “pointer arithmetic”.  
   (4 marks)
   
   b) Explain what you might expect to happen, and why it might happen, if the following code were to be executed.  
   (4 marks)
   
   ```c
   int my_array[10];
   int i;
   for (i=0 ; i< 100; i++)
       my_array[i]= i*2;
   ```

   c) Suppose you wish to construct a binary tree, which comprises a set of linked nodes, each of which can contain, as its data, a string of unlimited length. Suggest a suitable C data structure design for such a node, and explain why you have chosen such a design.  
   (4 marks)
   
   d) Now, illustrate the use of your data structure, by writing code to create an instance of a node, and to store the string “this exam is easier than I expected” in it.  
   (4 marks)
   
   e) Design a recursive function for traversing the binary tree, starting at the root of the tree, such that every node of the tree is visited only once. You may write your function either in pseudocode, or in C. Using a sample binary tree, illustrate the order in which the nodes in the tree are visited by the function.  
   (4 marks)
3. a) Explain what is meant by the terms **static data** and **dynamic data**, and, using an example real-world application of your choice, describe why a C programmer might choose to use dynamic memory.  

b) Examine the following code, and draw diagrams to show a snapshot of the state of the memory (for both static and dynamic data), by the time execution (which starts at `main()`) reaches the part of the code marked with the arrow. Make sure your diagrams are clear, and annotate them with comments to explain how the memory is being used.

```c
struct node {
    int data;
    struct node *next_ptr;
};

void f(struct node *p) {
    struct node *n_ptr;
    n_ptr = malloc(sizeof(struct node));
    p->next_ptr = n_ptr;
}

int main (int argc, char *argv[]) {
    struct node *list_ptr;
    list_ptr = malloc(sizeof(struct node));
    f(list_ptr);
    /* what is the state of memory now? */
}
```

(c) What is a “memory leak”? Why is it bad news for a programmer? Suggest two approaches that a programmer might take to help avoid memory leak.  

(d) Imagine you created have a linked list of `struct node` (defined in part (b), above) elements. You also have a `(struct node *) pointer ptr`, which can be set to point to any element in the list.

Write C code to create and insert a new node immediately **after** the node pointed to by `ptr`.  

Now change the definition of `struct node`, so that it can be used to create doubly-linked lists.  

Assuming that your list of elements is now doubly-linked, write C code to create and insert a new node immediately **before** the node pointed to by `ptr`.  

4. a) What is meant by ‘multiple inheritance’ in C++? How does Java provide similar functionality? Explain which of these, in your opinion, is better. (5 marks)

b) What is meant by a ‘polymorphic function’? How are these invoked? (2 marks)

c) How does polymorphism in C++ differ from that in Java? What is the performance implication of this difference? (3 marks)

d) Assume that your C++ compiler translates into plain C as an intermediate step. Explain, using examples and small fragments of C, how code for implementing virtual functions can be generated by the C++ compiler. (5 marks)

e) What is generic programming, and how is this provided for in C++? Give examples. How does Java provide similar functionality? Discuss the pros and cons of these different approaches. (5 marks)