UNIVERSITY OF DUNDEE

BSc EXAMINATION
School of Computing
2 hours

December 2010

Applied ComputingAC2A (AC21003)

This paper contains four questions.
Answer ALL questions.

Only calculators approved by the School for exam use may be used in this exam.

This paper contains four questions.

1. **(a)** Why is C++ described as a multi-paradigm programming language? In your answer you should consider the different approaches to problem solving that C++ supports and explain or otherwise give examples of why each approach is useful. **[6 marks]**

(b) Explain the purpose of each line in the following C++ program:

```
1.  // A small C++ program
2.  #include <iostream>
3.
4.  int main()
5.  {
6.      std::cout << Hello World << std::endl;
7.      return 0;
8.  }
```

[10 marks]

(c) Alter the C++ program from part (a) to do the following:

(i) Print a message asking the user to type in their full name.

(ii) Read the user's input into a string.

(iii) Print out a personalised greeting for the user which incorporates the supplied name.

[9 marks]

2. **(a)** What is the main purpose of a constructor in a C++ class? Your answer should explain under what circumstances a constructor would be called and what effect it has on the memory allocation for an object. **[12 marks]**

   (b) For the C++ class declaration given below, write a constructor for a GradeBook class that uses an initialisation list and allows the programmer to specify the following:

   - A string for the courseName
   - An integer for the courseID
   - A default value of 10 for numStudents
   - A dynamically allocated array of size numStudents

```
// GradeBook.h
#include <string>

class GradeBook
{
  private:
    std::string courseName;
    int courseID;
    std::string *students;
    int numStudents;
  public:
}
```

[13 marks]

This paper contains four questions.

3. **(a)** Explain the terms "*passing by value*" and "*passing by reference*" in the context of passing an object (such as an instance of a class) to a method. Also within this context, explain the terms "*deep copy*" and "*shallow copy*" and identify the situations in which a shallow copy can cause problems. What C++ programming feature is needed to overcome these problems? Give an outline of how this feature solves the problems. **[12 marks]**

(b) Explain the purpose of a destructor method. **[5 marks]**

(c) What is the main purpose of a template class in C++? Give an example of a situation where a template class would be useful and describe the advantages and disadvantages of using the template in that situation. **[8 marks]**

4. A bank wishes to set up a computer system to keep track of the three types of accounts it offers. The requirements for this software are as follows:

- Account type 1: The simplest account offers a customer only a facility to store money. No interest is paid, and the customer can only view the balance, deposit and withdraw cash without a fee.
- Account type 2: The savings account extends the basic account by paying interest at a rate that may change from time to time. The savings account offers the same deposit facility as the simple account, but a withdrawal is only allowed upon payment of a fixed fee.
- Account type 3: A cheque account extends the savings account in that it also pays interest (though usually at a different rate than the savings account) but also offers two methods of withdrawing money. The customer may withdraw cash in the same way as with the savings account, or they may cash a cheque.

(a) Use inheritance to create an object-oriented design to satisfy the above requirements. The design should be presented by showing each class with a name and a list of its data fields and methods, and stating which class(es) inherit which other class(es). You may use C++ code to illustrate your answer if you prefer. **[10 marks]**

(b) Explain why inheritance is suitable for this design, why you have given each class the data fields and methods it has and why the inheritance structure you have used is appropriate here. **[8 marks]**

(c) Write some C++ code that shows how 5 account objects of various types could be created and stored in a std::vector collection. **[7 marks]**

End of examination paper