OLLSCOIL NA hÉIREANN

National University of Ireland, Galway

SPRING EXAMINATIONS 1999/2000

Third University Examination in Information Technology

CT305 ADVANCED PROGRAMMING TECHNIQUES

Prof. D. Bell

Dr. G. Lyons

Mr. K. Power

Time Allowed: **3 Hours**

Answer five (5) questions in total.
Answer at least two (2) questions from each section.
All questions carry equal marks.
Please use a separate answer book for each section.

Section A – Object-Oriented Programming and Software Design

Q1 Object-Oriented Programming

(a) Briefly describe Encapsulation, Inheritance, and Polymorphism.

(b) Describe two forms of polymorphism in object-oriented programs.

(c) How is polymorphism related to dynamic binding?

(d) Write a class `PhoneNumber`. Decide what data members should be part of the class. Write a constructor, destructor, copy constructor, and overloaded assignment operator. Also overload the stream insertion operator and the stream extraction operator to allow input and output of `PhoneNumber` objects. Show the class (header file) and also the implementation of all methods (implementation file). Write a simple `main()` function that show how to create at least three `PhoneNumber` objects. Show the use of the operators that you overloaded.

p.t.o.

Q2 Programming Language Idioms and Design Patterns

(a) What is a design pattern?

(b) Explain the Abstract Factory design pattern, using an example to illustrate your answer.

(c) Explain the Open/Closed Principle, using an example as appropriate.

(d) Briefly discuss the Counted Body Idiom. Draw a sketch illustrating the Counted Body Idiom.

Q3 Generic Programming

(a) What is the STL?

(b) What does generic programming mean?

(c) List the essential components of the STL, and briefly discuss how they work together.

(d) Write a `Queue` template class (header file only).

(e) Discuss, using examples, the different idioms for using the vector class from the C++ standard library.

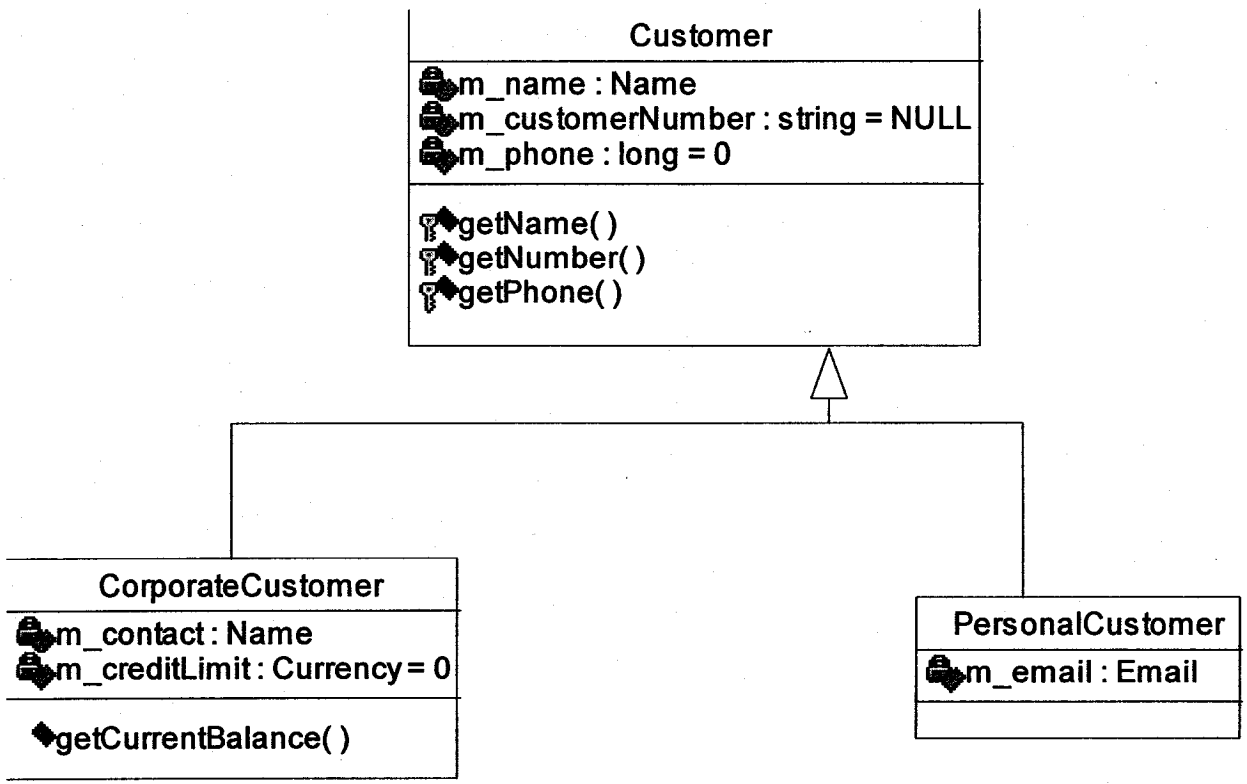
Q4 Object-Oriented Programming

(a) Given the UML class diagram on the next page, write the C++ implementation.

- Class Customer is an abstract base class.
- You need only show the header files and the implementation files.
- You should use Orthodox Canonical Form where appropriate.

(b) What is an abstract base class?

(c) Explain the C++ exception-handling mechanism.


Customer
m_name : Name
m_customerNumber : string = NULL
m_phone : long = 0
getName( )
getNumber( )
getPhone( )
CorporateCustomer
m_contact : Name
m_creditLimit : Currency = 0
getCurrentBalance( )
PersonalCustomer
m_email : Email

Section B – Algorithms and Data Structures

Q5 Graph Theory

(a) Explain the naïve implementation of the `UNION` and `FIND` algorithms for graphs. What are the disadvantages of this approach? Explain the techniques of height balancing and path compression that are used for improving the performance of these operations.

(b) Given the following Arc List, use the `UFSet` algorithm to determine whether the graph is connected. Show your work step-by-step. Draw the graph represented by the Arc List.

Arc List

1	5
1	8
2	4
3	8
3	7
5	9
6	2

(c) Given the graph from part (b), show how the graph can be represented using

(i) an Adjacency Matrix, and

(ii) an Adjacency List.

What are the disadvantages of using an Adjacency Matrix?

Q6 Searching Strategies

(a) Explain how a state space is represented by the four-tuple (N, A, S, GD). You should explain exactly what N, A, S, and GD are in this context. In this context, what is a solution path?

(b) Discuss the implementation issues involved with the development of both algorithms. What factors need to be considered when choosing between depth-first search and breadth-first search?

(c) Give the algorithm for depth-first search.

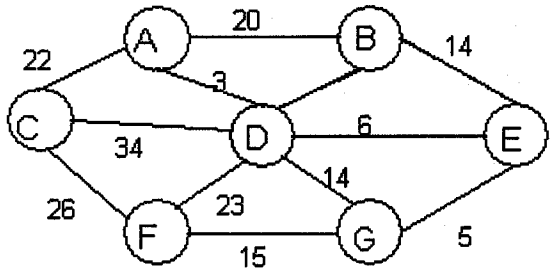
Q7 Minimum Spanning Trees

(a) What is a Spanning Tree? What is the MST Property? What are the 2 fundamental MST algorithm types?

(b) How does Kruskal's Fast Algorithm improve on Kruskal's Basic Algorithm?

(c) Explain Prim's Algorithm for MST. How does it differ from Kruskal's Algorithm?

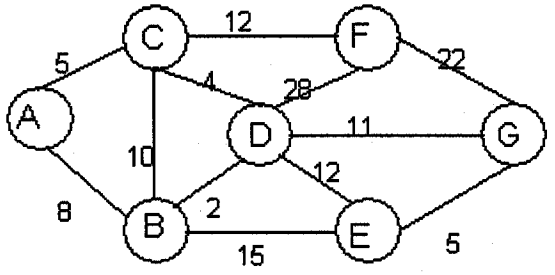
(d) For the following graph, show the step-by-step implementation of Kruskal's Fast Algorithm. Hence find the MST.



Q8 Shortest Path Algorithms

(a) Discuss the goals and problems involved in developing shortest path algorithms.

(b) Show how Dijkstra's Algorithm for finding the shortest path tree works, using the following graph as an example. Show each intermediate step.



(c) Describe Tarjan's implementation of Dijkstra's shortest path algorithm.