GX1308

OLLSCOIL NA hÉIREANN

National University of Ireland, Galway

SPRING EXAMINATIONS 1998/1999

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 Generic Programming

(a) What is the STL? (3 marks)

(b) What does generic programming mean? (2 marks)

(c) List the essential components of the STL, and briefly discuss how they work together. (4 marks)

(d) With regard to iterators, what is meant by control abstraction? (1 marks)

(e) Write the general form of a class template. (2 marks)

(f) Write a binary tree template class. (5 marks)

(g) Compare and contrast between templates and overloading. (3 marks)

Q2 Programming Language Idioms and Design Patterns

(a) What is a design pattern? (3 marks)

(b) Explain any **one (1)** of the following design patterns, using examples as appropriate. (5 marks)
Abstract Factory
Singleton
Proxy

(c) Briefly discuss the Handle/Body Idiom. Draw a sketch illustrating the Handle/Body Idiom. Write a simple example that illustrates this idiom. (5 marks)

(d) Briefly discuss the Counted Body Idiom. Draw a sketch illustrating the Counted Body Idiom. (5 marks)

(e) Distinguish between the two language idioms associated with the use of C++ vectors. Give a simple example of each. (2 marks)

Q3 Object-Oriented Programming

(a) With respect to software development, briefly discuss the importance of separating interface from implementation details. (3 marks)

(b) Encapsulation, Inheritance, and Polymorphism are often called "The Three Pillars of Object-Oriented Programming". (7 marks)

Briefly discuss these three important concepts. In your discussion, mention the different ways of implementing polymorphism.

(c) Briefly discuss the relationship between polymorphism and dynamic binding. (3 marks)

(d) Discuss operator overloading. Briefly mention any advantages or disadvantages. (2 marks)

(e) Given the following outline of a class Point, overload the addition operator (as a member function), the equality operator (as a member function), and the stream insertion operator (as a non-member function). (5 marks)

Note: You need only show the implementation, not the modified class interface.

```
class Point {
public: // public interface

  // --- accessors
  int x() const ;
  int y() const ;

  void draw () const ;

  // --- mutators
  void x(int newX) ;
  void y(int newY) ;

public:  // boilerplate

           Point () ;
           Point (int newX, int newY) ;
  virtual ~Point () ;

protected:

private:

  int    m_x ;
  int    m_y ;
} ;
```

Q4 Object-Oriented Programming Principles

(a) Explain the Open/Closed Principle, using an example as appropriate. (6 marks)

(b) Explain the Liskov Substitution Principle, using an example as appropriate. (6 marks)

(c) Explain the Dependency Inversion Principle, using an example as appropriate. (6 marks)

(d) What is the difference between a class library and a framework? (2 marks)

Q5 Object-Oriented Programming

(a) Given the UML class diagram on the next page, write the C++ implementation. (10 marks)

NOTES: You need only show the header files, not the source files.

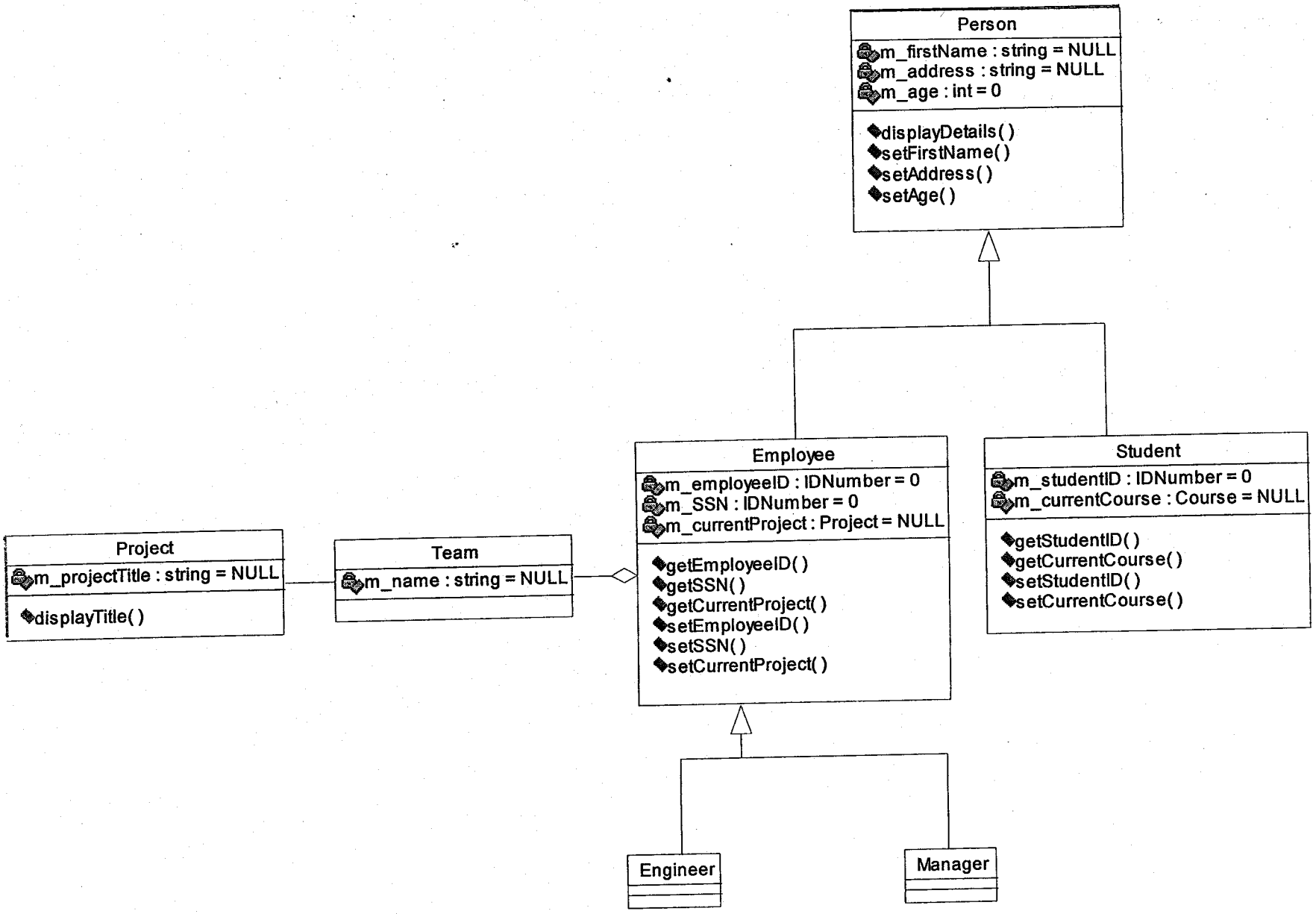
You should use Orthodox Canonical Form where appropriate.

(b) What is an abstract base class? (2 marks)

(c) Outline an abstract base class `Order` that contains the following methods: (4 marks)

```
calculateTotal()
display()
print()
```

(d) Briefly explain the C++ exception-handling mechanism. (4 marks)


Person
m_firstName : string = NULL
m_address : string = NULL
m_age : int = 0
displayDetails( )
setFirstName( )
setAddress( )
setAge( )
Employee
m_employeeID : IDNumber = 0
m_SSN : IDNumber = 0
m_currentProject : Project = NULL
getEmployeeID( )
getSSN( )
getCurrentProject( )
setEmployeeID( )
setSSN( )
setCurrentProject( )
Student
m_studentID : IDNumber = 0
m_currentCourse : Course = NULL
getStudentID( )
getCurrentCourse( )
setStudentID( )
setCurrentCourse( )
Project
m_projectTitle : string = NULL
displayTitle( )
Team
m_name : string = NULL
Engineer
Manager

Section B – Algorithms and Data Structures

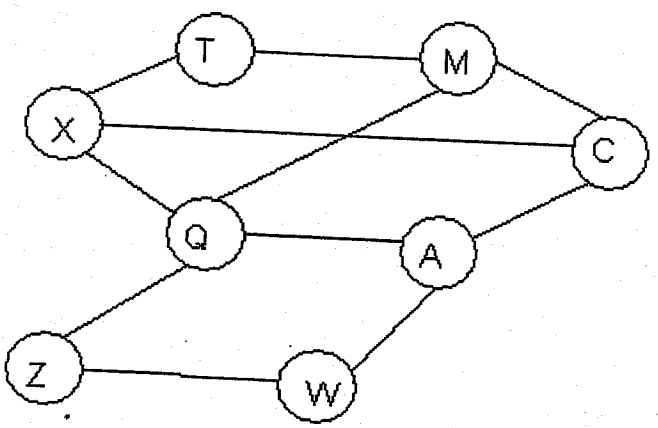
Q6 Graph Theory

(a) Given the following graph, show how the graph can be represented using (6 marks)

(i) an Adjacency Matrix, and

(ii) an Adjacency List.

Mention the disadvantages, if any, of using an Adjacency Matrix. How does an Adjacency List improve on an Adjacency Matrix?



(b) UFSet is a way of representing graphs using sets. Explain the naïve implementation of the UNION and FIND operations for graphs. What are the disadvantages of this approach? Explain two techniques for improving the performance of these operations. (7 marks)

(c) 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. (7 marks)

Arc List

4 2

2 3

3 7

7 6

9 6

9 3

Q7 Searching Strategies

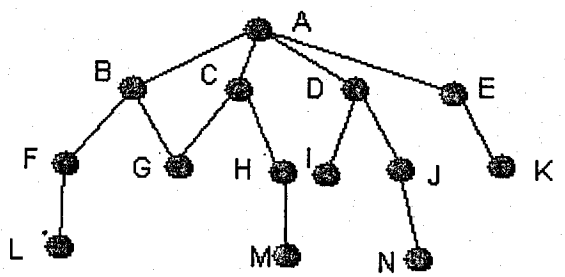
(a) Explain how a state space is represented by the four-tuple (N, A, S, GD). In this context, what is a solution path? (4 marks)

(b) There are two fundamental possibilities when searching a graph: breadth-first search and depth-first search. Briefly discuss the implementation of both algorithms. What factors need to be considered when choosing between depth-first search and breadth-first search? (6 marks)

(c) Give the algorithm for depth-first search. (5 marks)

(d) What needs to be changed in the algorithm for breadth-first search? (1 mark)

(e) Using the following graph as an example, show a trace of the depth-first search algorithm. You need only show the first eight (8) steps. (4 marks)



Q8 Minimum Spanning Trees

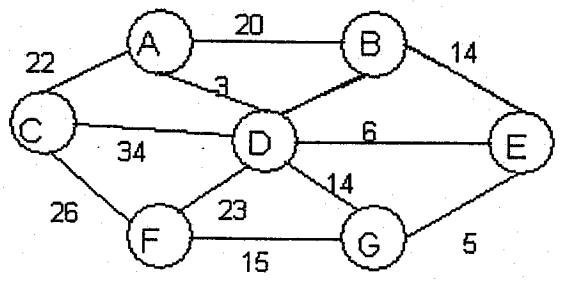
(a) What is a Spanning Tree? What is the MST Property? What are the 2 fundamental MST algorithm types? (5 marks)

(b) How does Kruskal's Fast Algorithm improve on Kruskal's Basic Algorithm? (3 marks)

(c) Explain Prim's Algorithm for MST. (3 marks)

(d) What is the main difference between Kruskal's Algorithm and Prim's Algorithm? (2 marks)

(e) For the following graph, show the step-by-step implementation of Kruskal's Basic Algorithm and Kruskal's Fast Algorithm. Hence find the MST. (7 marks)

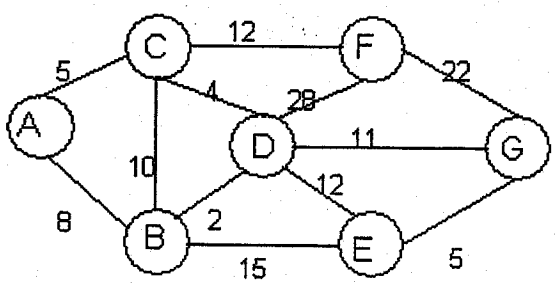


Q9 Shortest Path Algorithms

(a) Discuss the goals and problems involved in developing shortest path algorithms. (5 marks)

(b) Define a Heap data structure and show by means of example how the `upheap` and `downheap` operations work. (5 marks)

(c) Show how Dijkstra's Algorithm for finding the shortest path tree works, using the following graph as an example. Show each intermediate step. (7 marks)



(d) Tarjan's implementation of Dijkstra's shortest path algorithm offers an improvement on Dijkstra's original algorithm. Briefly discuss this improvement. (3 marks)