Maynooth University
National University of Ireland Maynooth

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

JANUARY 2017 EXAMINATION

CS264

Software Design

Dr. S. Flynn, Dr. A. Winstanley, Dr. J. McDonald

Time allowed: 2 hours

Answer at least three questions
Your mark will be based on your best ***three*** answers

All questions carry equal marks

[25 marks]

1 (a) Write a function in C++ that takes as input a positive integer n and returns as output an array of integers of length n, where the array is populated by the sequence 0..n-1. [8 marks]

(b) A given C++ library maintains a list of customer names in a linked list data structure. The list is made of `customers_nodes` where each `customer_node` stores a customer's `name` and a pointer to the `next customer_node` in the list. [9 marks]

Provide a definition of a C++ struct for the `customer_node` type in described above.

Provide definitions for a C++ function called `find_customer` which takes as input a pointer to the first `customer_node` in the list and a customer name for which to search. The function should traverse the list, searching for the presence of a `customer_node` containing the specified name. The function should return `true` if it finds a node containing the search name or should return `false` otherwise. The function should have the prototype :

```
bool find_customer(customer_node *head, string name);
```

(c) The C++ code below provides a class definition for a 2D Vector class that is to be used by a linear algebra library. [8 marks]

```
class Vector{
public:
     Vector(int x, int y): x_(x), y_(y) {};
     Vector(int n): x_(n), y_(n) {};

protected:
     int x_, y_;
}
```

Demonstrate how C++ templates can be used to convert the above class to a generic implementation i.e. where the internal type used to represent the vector components is set by a template parameter.

[25 marks]

2 (a) The code below shows a set of C++ classes and a main function that instantiates a variable of type B. Provide a UML class diagram to illustrate the relationships between the four classes. [10 marks]

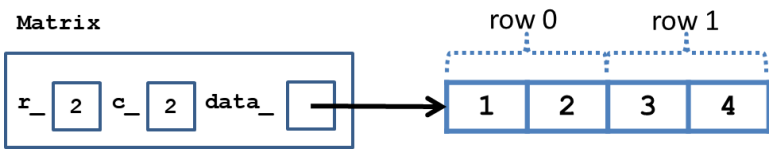
What will the output of the program be upon execution? Explain your answer.

```
#include <iostream>

using namespace std;

class C{
public:
  C(){ cout << "C\n"; };
  virtual ~C() { cout << "~C\n"; };
};

class D{
public:
  D(){ cout << "D\n"; };
  virtual ~D() { cout << "~D\n"; };
};

class A{
public:
  A(){ cout << "A\n"; };
  virtual ~A() { cout << "~A\n"; };
protected:
  C c;
};

class B: public A{
public:
  B(){ cout << "B\n"; };
  virtual ~B(){ cout << "~B\n"; };
protected:
  D d;
};

int main(){
  B b;
}
```

(b) The C++ code below defines a `Matrix` class for storing a 2D matrix of numbers for use in a linear algebra library. The class provides a constructor that takes the number of rows and columns of the matrix as input, and initialises a matrix of zeros of that size. The matrix data is stored row-by-row in a 1D linear block of memory whose address is stored in the `data` member variable. For example the matrix: [8 marks]

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

would be stored as:




```
class Matrix{
public:
  Matrix(int row, int col):
    r_(row),c_(col),data(new int[r_*c_]){
    // initialise the matrix elements to zero
    memset(data, 0, r*c*sizeof(int));
  };
  virtual ~Matrix(){ delete[] data; };

  virtual void set(int row, int col, int value;

  void print();

protected:
  int r_, c_; // number of rows and columns
  int *data; // pointer to matrix data
};
```

Complete the above class by providing definitions for the `set` and `print` methods. The `set` method should take as input a `row` and `column` index and a `value` which it should store at that location in the matrix. The `print` method should print out the contents of the matrix, row-by-row, with each row on a separate line.

(c) A symmetric matrix is a **square matrix** where the for any given (row,col) location, the value at that location is equal to the value at location (col,row). The following is an example of a symmetric matrix: [7 marks]

$$\begin{bmatrix} 2 & 3 & 5 \\ 3 & 1 & 6 \\ 5 & 6 & 7 \end{bmatrix}$$

Derive a `SymMatrix` class from the `Matrix` class defined in part (b). The new class should include the following methods:

- `SymMatrix(int r)` : a single parameter constructor that creates a square matrix i.e. with `r` rows and `r` columns.
- `set(int row, int col, int value)` : that should set the values at `(row,col)` and `(col,row)` to `value` i.e. to enforce the symmetry constraint.

Through the use of example code, explain the need for the `virtual` keyword in the declaration of the `set` method in the `Matrix` class.

[25 marks]

3 (a) A local software company has been employed to develop a new software system for the local county council office, called *ParcelPlanner*. The software system will be used to assist council planners to track ownership of parcels (i.e. regions or sites) of land in the county. [15 marks]

Each region of land will be represented using a `Parcel` class in the system, where each parcel will have an `Owner`, `ZoneID`, and geographic coordinate (`GeoCoord`). The County will be represented using a `ParcelDB` class which will maintain a database of all the known `Parcel`s for that county. `ParcelDB` will provide an interface for adding a list of `Parcel`s to the database and searching for parcels. The search functionality is covered in part (b) of this question.

A `SystemInterface` class will provide a frontend for the `ParcelDB` by providing methods for inputting parcel data from a planner's smartphone or from a spreadsheet file. To do this the system will define an abstract `ParcelInputter` class which will define a `GetNextParcel` method which will retrieve the next `Parcel` from the given input. A `SmartPhone` class and a `SpreadSheet` class will both provide specialisations of the `ParcelInputter` class to handle the specifics of each input type.

To input `Parcel`s to the system the `SystemInterface` class will implement a `SmartphoneInput` method and a `SpreadSheetInput` method. Each of these will construct the appropriate concrete class which will be passed (as a `ParcelInputter` class) to the `ParcelDB`. The `ParcelDB` will then iterate through the `ParcelInputter` through repeated calls to `GetNextParcel`.

Given the above system description, provide a UML class diagram that illustrates the structure of the system. Note that you should include all of the relevant classes, associations, methods/operations, and attributes.

(b) To make the system as extensible as possible the designers will use the strategy pattern for implementing the search functionality. Here `ParcelDB` class will have a member variable of type `SearchCriteria`, where `SearchCriteria` defines a `CheckParcel` method. This method will take as input a `Parcel` and will return `true` or `false` depending on whether the parcel [10 marks]

meets the search criteria. Three concrete search criteria classes will be implemented `ZoneCriteria`, `GeographicCriteria`, `SizeCriteria`.

To search the system the `SystemInterface` class will define `ZoneSearch`, `GeographicSearch`, and `SizeSearch` methods each of which will construct an appropriate concrete `SearchCriteria` class which it will then set as the `ParcelDB`'s current `SearchCriteria`.

Using your solution to part (a), provide the new classes and associations required to provide for the extra functionality and system structure defined above. Explain how this design is an instance of the strategy pattern.

[25 marks]

4 (a) Explain what the intent of the singleton design pattern is and show how it can be implemented in C++. Explain the elements of the C++ implementation that ensure that the resulting class is a singleton. [8 marks]

(b) Explain the intent and operation of the observer pattern? Provide a UML diagram of the observer pattern. Explain its overall structure and the role of each of its elements. [9 marks]

(c) Explain the model-view-controller approach to structuring an application's design. In your opinion is the observer pattern relevant to this approach, and if so how? [8 marks]