UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

PROGRAMMING TECHNIQUES II

COURSE CODE – EE272

SUPPLIMENTARY EXAMINATION

JULY 2011

DURATION OF THE EXAMINATION - 3 HOURS

INSTRUCTIONS TO CANDIDATES

1. There are **FIVE** questions in this paper. Answer any **FOUR** questions only.
2. Each question carries equal marks.
3. Show all your steps clearly in any calculations.
4. State clearly any assumptions made.
5. Start each new question on a fresh page.

DO NOT OPEN THIS PAPER UNTIL PERMISSION HAS BEEN GIVEN BY THE INVIGILATOR.

Question 1

a) How is it that polymorphism enables programming "in the general" rather that "in the specific"? Discuss two advantages of programming "in the general". [4]

b) Discuss two problems of programming with the `switch` logic. Using an example, explain how polymorphism can be an effective alternative to switch logic. [4]

c) Information hiding is one of the key features that distinguish object-oriented programming from structured programming. Using an example, explain the rationale of information hiding and how it relates to the following object-oriented programming concepts: *abstraction*, *coupling*, and *cohesion.* [4]

d) Discuss the ways in which inheritance promotes software reuse, saves time during program development and helps prevent errors. [4]

e) Describe the ways by which a derived class may inherit from a base class. [5]

f) Using an example, explain the relationship between function templates and function overloading? [4]

Question 2

a) Explain how does polymorphism promote extensibility of software design? [4]

b) A bank requires a software system for managing customers' bank accounts. All customers at this bank can deposit into and withdraw from their accounts. The bank has two types of accounts: savings and current. A savings account earns interest on the money held meanwhile a current account charges a fee per transaction. All accounts have methods for crediting, debiting, and retrieving account balances. Savings accounts keep information on interest rates and provide a way of computing interest earned. On the other hand current accounts keep information on the fee charged per transaction. Current accounts refine the general functionality of crediting and debiting all accounts so that whenever a transaction is performed successfully a transaction fee is deducted from the account balance.

From the description of the proposed banking application problem above, give a detailed account of *why* and *how* the programming techniques of *inheritance* and *polymorphism* could be especially effective for solving a problem of this nature. [21]

Question 3

Package-delivery services, such as FedEx, DHL, and UPS, offer a number of different shipping options, each with specific costs associated.

Create an inheritance hierarchy in the form of a class diagram to represent the various types of packages. Use `Package` as the base class of the hierarchy, then include classes `TwoDayPackage` and Overnight that derive from `Package`. Base class `Package` should include data members representing the *name*, *address*, *city*, and *region* for both the sender and the recipient of the package, in addition to data members that store the *weight* (in kilograms) and *cost per kilogram* to ship the package. `Package`'s constructor should initialise these data members. Ensure that the weight and cost per kilogram contain positive values.

`Package` should provide a public member function `calculateCost` that returns a double indicating the cost associated with shipping the package. `Package`'s `calculateCost` function should determine the cost by multiplying the weight by the cost per kilogram. Derived class `TwoDayPackage` should inherit the functionality of base class `Package`, but also include a data member that represents a flat fee that the shipping company charges for two-day-delivery service. `TwoDayPackage`'s constructor should receive a value to initialise this data member. `TwoDayPackage` should redefine member function `calculateCost` so that it computes the shipping cost by adding the flat fee to the weight-based cost calculated by base class `Package`'s `calculateCost` function.

Class `OverNightPackage` should inherit directly from class `Package` and contain an additional data member representing and additional fee per kilogram charged for overnight-delivery service. `OverNightPackage` should redefine member function `calculateCost` so that it adds the additional fee per kilogram to the standard cost per kilogram before calculating the shipping cost.

(i) Draw a class diagram depicting the three classes and their relationship. [3]
(ii) Write the C++ interface of each class. [6]
(iii) Write the C++ implementation of each class. [12]
(iv) Write a C++ program that creates objects of each type of package and tests their member function `calculateCost`. [4]

Question 4

Analyse the following programs and determine their outputs. Show all working.

(a)

```
#include "stdafx.h"
#include <iostream>

using namespace std;
using namespace System;

int main(array<System::String ^> ^args){
	int i, j, c = 9, m, k;
	for (i = 1; i <= 5; i++) {
		for (k = 1; k <= c; k++) {
			printf(" ");
		}
		for (j = 1; j <= i; j++) {
			printf("%2d", j);
		}
```

```
            for (m = j - 2; m > 0; m--) {
                  printf("%2d", m);
            }
            printf("\n");
            c = c - 2;
      }
      Console::ReadKey();
    return 0;
}
```

[9]

(b)

```
#include "stdafx.h"
#include <iostream>

using namespace std;
using namespace System;

int main(array<System::String ^> ^args){

      char prnt = '*';
      int i, j, k, s, nos = -1;

      for (i = 5; i >= 1; i--) {
            for (j = 1; j <= i; j++) {
                  printf("%2c", prnt);
            }
            for (s = nos; s >= 1; s--) {
                  printf("  ");
            }
            for (k = 1; k <= i; k++) {
                  if (i == 5 && k == 5) {
                  continue;
                  }
                  printf("%2c", prnt);
            }
            nos = nos + 2;
            Console::WriteLine();
      }
      nos = 5;
      for (i = 2; i <= 5; i++) {
            for (j = 1; j <= i; j++) {
                  printf("%2c", prnt);
            }
            for (s = nos; s >= 1; s--) {
                  printf("  ");
            }
            for (k = 1; k <= i; k++) {
                  if (i == 5 && k == 5) {
                        break;
                  }
                  printf("%2c", prnt);
            }
            nos = nos - 2;
            Console::WriteLine();
      }
      Console::ReadKey();
    return 0;
}
```

[16]

Question 5

Figure 5 depicts an intersection between two roads. Traffic through Road 1 is one way and in the direction shown by the arrow labeled *Traffic 1*. This traffic is controlled by traffic light *TB*. Traffic through Lane A flows in the direction shown by the arrow

labeled *Traffic 2*. This traffic is controlled by traffic light *TA*. Each traffic light module has three lights (RED, GREEN, and AMBER). Traffic through Lane B flows in the direction shown by the arrow labeled Traffic 3, and is NOT regulated by any traffic light.

The traffic control lights for controlling Traffic 1 and Traffic 2 are required to operate so that traffic flows as follows: Traffic 1 is allowed 1 minute to enter Lane A, while Traffic 2 is allowed 2 minutes. Traffic 1 is allowed to pass through the bridge when the GREEN light in TB is ON, otherwise the Traffic 1 is stopped. Similarly, Traffic 2 can only pass through Lane A when the GREEN lamp in TA is on. When a traffic control light turns from GREEN to RED, for 3 seconds it flashes the AMBER light and then switches ON the RED lamp. Assume that, initially (i.e. before the control regime come into effect), all traffic is stopped for 15 seconds.

You are required to design and code a well documented and indented C++ program that when executed would make sure that the Control Lights TA and TB control the traffic through the road intersection according to the above control regime described above.

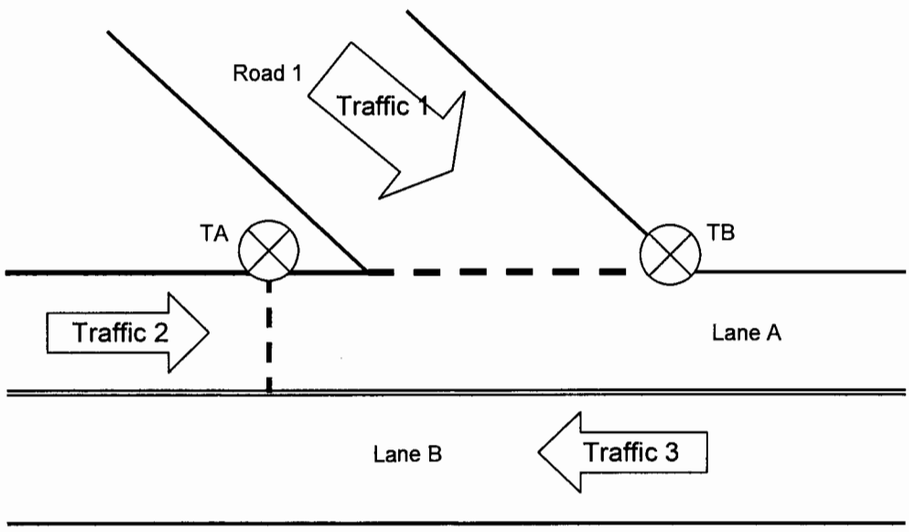

Figure 5. Road intersection for question 5

Assume that the computer in which the program will be executed works in the *hexadecimal number system* while the circuit interfacing to the control lights works in *binary format*. Also, when writing your program, assume that the following two functions have been defined as members of a library class called 'TrafficControl.h':

```
void sendControl(int p, int g, int a, int r)
void delay(int seconds)
```

Function `sendControl()` sends control data to turn ON and OFF the lamps of a given traffic light module. It takes four arguments: `p, g, a,` and `r`. Argument `p` is a integer representing the address of the port where control data for a given traffic module is to be sent. The port numbers are 1 and 2, for Traffic Lights Modules TA and TB, respectively. Arguments `g, a,` and `r` are integer values for controlling the state of the green, amber, and red lamps, respectively. For example, if the value of `g = 1`, this turns-on the green lamp and 0 turns it off. The function `delay()` delays execution of the next statement in a program for a time period equivalent to the seconds supplied as an argument.

State clearly any other assumptions you have made in your solution.

As part of your solution produce the following artefacts:

(i) A state chart modeling the behaviour of the traffic control system. [4]

(ii) A table that shows the sequence of how the six lamps would be switched ON and OFF. [5]

(iii) Using appropriate identifiers, write a C++ program based on the state chart in (i) and the table in (ii) to meet the requirements of the traffic lights control regime. Your code should include a class, called '`ControlLight.h`', for representing a single traffic light module. In addition to the control program, your code must show clearly the interface of class '`ControlLight.h`' (with appropriate data members and functions) and its implementation. [16]

END OF PAPER