UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE & ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

MAIN EXAMINATION

May 2018

TITLE OF PAPER: PROGRAMMING TECHNIQUES II

COURSE CODE: EEE272/EE272

DURATION: 3 HOURS

INSTRUCTIONS:

1. There are five (5) questions in this paper. Answer question 1 and any other three (3) questions.
2. Each question carries 25 marks.
3. Use correct notation and show all your steps clearly in any program analysis.
4. All programs should be sufficiently commented and indented for clarity.
5. Start each question in a new page.

This paper should not be opened until permission has been given by the invigilator.

This paper contains eight (8) pages including this page.

Question 1

a. Explain the meaning of the `inline` keyword in C++ function declarations. What is a potential drawback of using it? [3]

b. What is the meaning of the following keywords? Use short program segments to complement your explanation. [4]

i. `static`

ii. `const`

c. How are overloaded functions differentiated from each other when a program is executed? [2]

d. Explain why it is sometimes useful to overload a constructor. [2]

e. Define the principles of encapsulation and information hiding in object oriented programming. How does the definition of classes promote these principles? [4]

f. Explain why a class might provide set and get functions to update and access data members. [2]

g. What is the purpose of access specifiers? Explain the access control level provided by each access specifier. [6]

h. By using an example explain the difference between a base class and a derived class. [2]

Question 2

Analyze the following programs, and determine their exact outputs.

a. [4]

```
#include <iostream>
using namespace std;

class Box
{
      double width;

      public:
            friend void printWidth(Box box);
            void setWidth(double wid);

} ;

void Box::setWidth(double wid)
{
      width = wid;
}
```

```
void printWidth(Box box)
{
      box.width = box.width * 2;
      cout « "Width of box: " « box.width « endl;
}

int main()
{
      Box box;
      box.setWidth(10.0);
      printWidth(box);
      return 0;
}
```

b. [6]

```
#include <iostream>
using namespace std;

class Base{

public:
      Base(){ cout<<"Base"<<endl; }
      Base(int i){ cout<<"Base"<< i <<endl; }
      ~Base(){ cout<<"Destruct Base"<<endl; }
};

class Derived: public Base{

public:
      Derived(){ cout<<"Derived"<<endl; }
      Derived(int i):Base(i){ cout<<"Derived"<< i <<endl; }
      ~Derived(){ cout<<"Destruct Derived"<<endl; }
};

int main(){

      Base a;
      Derived d(2);
      return 0;
}
```

c. [4]

```
#include <iostream>
using namespace std;

class C
{
      public:
            virtual string toString()
            {
                  return "C";
            }
};

class B: public C
{
    string toString()
      {
```

```
        return ": ";
    }
};

class A: public B
{
    string toString()
      {
        return "A";
    }
};

void displayObject(C p)
{
    cout << p.toString();
}

int main()
{
    displayObject(A());
    displayObject(B());
    displayObject(C());
    return 0;
}
```

d. Write a short program that may be used by the swap function shown below. The program must display two values before and after they have been swapped. [7]

```
template<typename T>
void swap(T &var1, T &var2)
{
    T temp = var1;
    var1 = var2;
    var2 = temp;
}
```

f. Show the output of the following program, if you enter 1 0, what is the output of the following code? [4]

```
#include <iostream>
using namespace std;

int main()
{
  // Read two integers
  cout << "Enter two integers: ";
  int number1, number2;
  cin >> number1 >> number2;

  try
  {
    if (number2 == 0)
      throw number1;

    cout << number1 << " / " << number2 << " is "
      << (number1 / number2) << endl;
  }
  catch (int e)
```

```
    {
      cout << "Exception: an integer " << a <<
        " cannot be divided by zero" << endl;
    }

    cout << "Execution continues ..." << endl;

    return 0;
}
```

Question 3

a. What is inheritance in object oriented programming? Discuss ways by which it contributes to software reuse and short turnaround times in program development. [6]

b. Consider the following class definitions, then answer the questions that follow.

```
class TVGame
{
protected:
      string host;
      string game;
public:
      TVGame(string h, string g);
      GoToCommercial();
      // add the StartGame function here
};

class Jeopardy : public TVGame
{
private:
      int score;
public:
      Jeopardy(string h, string g, int s);
      // add StartGame function here
};
```

i. Write the function body for the constructor of Jeopardy. [2]

ii. Write the function header for constructor of Jeopardy as it would appear in the implementation file, Jeopardy.cpp. [2]

iii. Assume a class is to be derived from Jeopardy, e.g., JuniorJeopardy. The new class should inherit all data members. Which key word in the Jeopardy class would need to change to make this possible? [1]

c. Identify all errors in the following class definition. and then explain how to correct them.

[2]

```
class Example
{
      public:
            Example(int y = 10): data(y)
            {
                  // empty
            }

            int getIncrementedData() const
            {
                  return ++data;
            }

            static int getCount()
            {
                  cout<<"Data is "<< data << endl;
            }

      private:
            int data;
            static int count;
};
```

d. In the class it was mentioned that an operator+ for an expression such as `10 + Simple(3)` cannot be overloaded as a member function. Why is it so? Write a simple code segment that includes a class with data members and member functions to illustrate how the operator could be appropriately overloaded for the given expression. [7]

e. Define the concept of friendship in object oriented programming. Some people are against this concept: Discuss the possible negative aspects of friendship. [4]

Question 4

Write a complete C++ program that reads in test scores and applies two different curves to them. The program should,

- Contain a base class ScoreBank with two private data members: an integer pointer for the scores and a float for the average. The class should contain a member function *EnterScores* which asks the user how many test scores are needed, allocates enough memory, and reads in the scores. The class should also contain a member function *CalcAverage* which stores the average of the entered scores in the private float data member. Also have an *Output* function that prints a list of test scores and the average.

- Derive from ScoreBank a class Curve1 which contains a member function *Curve*. *Curve* sets the average score to 75. Find out how far away from 75 the actual average is and then add this value to each test score. Overload the output member function to print, original scores and the curved scores as well as the original and new average.

- Derive from ScoreBank a class Curve2 which contains a member function *Curve*. *Curve* sets the high score to 100 and scales the rest of the scores accordingly. Find the high score (provide a friend function to Curve2, called *Maximum*, to get the high score) and divide all scores by this value. Then overload the *Output* function to print the original scores, the new scores, and the averages for both sets.

- Include all other data members and member functions as necessary.

Write a driver program for these classes that creates objects of type Curve1 and Curve2 and asks the user to input a list of test scores into each object and then runs the Output functions from each object

i. Write the C++ interface. [5]
ii. Write the C++ implementation. [15]
iii. Write a C++ driver program for the class [5]

Question 5.

Write a complete C++ program that uses classes Rectangle and Point. Class Rectangle only stores Cartesian coordinates of type Point for the four corners of the rectangle. The class must include a set function that does the following:

- Before assigning the sets of coordinates to data members, it must verify that they are in the first quadrant with no single x or y coordinate larger than 20.0.
- It must also verify that the supplied coordinates specify a rectangle.

Other member functions include:

- A constructor that uses the set function to initialise the coordinates.
- Member functions to calculate length, width, perimeter and area.
- A member function which determines whether the rectangle is a square.

i. Write the C++ interface. [5]

ii. Write the C++ implementation. [15]

iii. Write a C++ driver program for the class [5]

End of paper