



COLLEGE OF ENGINEERING, SCIENCE AND TECHNOLOGY

School of Electrical & Electronics Engineering

Trade Diploma in Electronics Engineering

EEE525 Engineering Software

SPECIAL EXAMINATION

Semester 2, 2015

Date: As per Exam Time Table

Time: As per Exam Time Table

Venue: As per Exam Timetable

Instructions to Students

1. You are allowed an extra ten (10) minutes of reading time during which you are NOT allowed to write.
2. Attempt ALL questions in this examination booklet
3. Write your answers in the answer booklet provided.
4. Write your Student ID number on each page used.
5. Begin each Section on a fresh page and use both sides of the answer sheet.
6. You may use calculators provided they are non-programmable.
7. Clearly number the questions in your answer paper in their correct sequence and write legibly. Show all working.
8. Attach any extra sheets used to your answer booklet securely with the string provided.

Special Examination**Section A (60 marks)****Question 1 (4 marks)**

- i) Name 6 Phases of C++ Programs.
- ii) Outline the description of Escape Sequence “\n” and “\t”.

Question 2 (6 marks)

- i) For the coding given in appendix A, outline the application of using “std::” in lines 2,3 and 4.
- ii) What will be displayed on output if you enter 22 and 12 as input in the program of appendix A.

Question 3 (5 marks)

- i) Define Object oriented programming.
- ii) Outline the Object-oriented analysis and design process.

Question 4 (10 marks)

- i) Define Structured programming and the rules for Structured programming
- ii) Identify and correct the errors in each of the following:
 - a)

```
if ( age >= 65 );  
    cout << "Age is greater than or equal to 65" << endl;  
else  
    cout << "Age is less than 65 << endl";
```
 - b)

```
if ( age ==> 65 )  
    cout << "Age is greater than or equal to 65" << endl;  
else;  
    cout << "Age is less than 65 << endl";
```

Question 5 (10 marks)

- i) In appendix B, what does the program do? What does the following program print?
- ii) Define function and Recursive functions

Please Turn Over

Special Examination**Question 6 (5 marks)**

State whether the following are true or false. If the answer is false, explain why.

- a) An array can store many different types of values.
- b) An array subscript should normally be of data type float.

Question 7 (5 marks)

State whether each of the following are true or false. If false, explain why.

- a) Base-class constructors are not inherited by derived classes.
- b) A *has-a* relationship is implemented via inheritance.
- c) A Car class has an *is-a* relationship with the SteeringWheel and Brakes classes.
- d) Inheritance encourages the reuse of proven high-quality software.
- e) When a derived-class object is destroyed, the destructors are called in the reverse order of the constructors.

Question 9 (5 marks)

- i) What are virtual functions? Describe a circumstance in which virtual functions would be appropriate.
- ii) Define Polymorphism.

Question 10 (10 marks)

- i) List five common examples of exceptions.
- ii) Give several reasons why exception-handling techniques should not be used for conventional program control.
- iii) Why are exceptions appropriate for dealing with errors produced by library functions?

Please Turn Over

Special Examination

Section B (40 marks)

Question 1 (5 marks)

Discuss, using your own experience, how drawing a structure/flow chart, as part of your preparation to coding, will help you with your final coding.

Question 2 (5 marks)

Write a C++ program to ask user to input two double floating (num1 and num2) numbers, perform addition and display the output.

Question 3 (10 marks)

A class of ten students took a quiz. The grades (integers in the range 0 to 100) for this quiz are available to you. You have to determine the class average on the quiz. Draw a flow chart for the example and write a C++ code using while loop as repetitions. The following Pseudocode can be used to draw flow chart and write C++ coding:

Set total to zero
Set grade counter to one
While grade counter is less than or equal to ten
 Input the next grade
 Add the grade into the total
 Add one to the grade counter
Set the class average to the total divided by ten
Print the class average

Please Turn Over

Special Examination

Question 4 (10 marks)

- i) Find the errors in each of the following program segments, and explain how the errors can be corrected.

```
a) int g()
{
    cout << "Inside function g" << endl;
    int h()
    {
        cout << "Inside function h" << endl;
    }
}
```

```
b) int sum( int x, int y )
{
    int result;
    result = x + y;
}
```

- ii) What does the program in appendix C do?

Question 5 (10 marks)

Write statements that perform the following tasks for an array called fractions:

- Define a constant integer variable arraySize initialized to 10.
- Declare an array with arraySize elements of type double, and initialize the elements to 0.
- Refer to array element 4.
- Assign the value 1.667 to array element 9.
- Assign the value 3.333 to the seventh element of the array.
- Print array elements 6 and 9 with two digits of precision to the right of the decimal point, and show the output that is actually displayed on the screen.
- Print all the array elements using a for statement. Define the integer variable i as a control variable for the loop. Show the output.

Please Turn Over

Special Examination

Appendix A

```
#include <iostream>
using std::cout; // program uses cout
using std::cin; // program uses cin
using std::endl; // program uses endl
int main()
{
    int num1; // first number to be read from user
    int num2; // second number to be read from user
    cout << "Enter two integers, and I will tell you the relationships they satisfy: ";
    cin >> num1 >> num2; // read two integers
    if ( num1 == num2 )
        cout << num1 << " is equal to " << num2 << endl;
    if ( num1 != num2 )
        cout << num1 << " is not equal to " << num2 << endl;
    if ( num1 < num2 )
        cout << num1 << " is less than " << num2 << endl;
    if ( num1 > num2 )
        cout << num1 << " is greater than " << num2 << endl;
    if ( num1 <= num2 )
        cout << num1 << " is less than or equal to " << num2 << endl;
    if ( num1 >= num2 )
        cout << num1 << " is greater than or equal to " << num2 << endl;
    return 0; // indicate that program ended successfully
} // end function main
```

Appendix B

```
#include <iostream>
using namespace std;
int main()
{
    int y; // declare y
    int x = 1; // initialize x
    int total = 0; // initialize total
    while ( x <= 10 ) // loop 10 times
    {
        y = x * x; // perform calculation
        cout << y << endl; // output result
        total += y; // add y to total
        x++; // increment counter x
    } // end while
    cout << "Total is " << total << endl; // display result
} // end main
```

Please Turn Over

Special Examination

Appendix C

```
#include <iostream>
using namespace std;
int mystery( int, int ); // function prototype
int main()
{
    int x, y;
    cout << "Enter two integers: ";
    cin >> x >> y;
    cout << "The result is " << mystery( x, y ) << endl;
} // end main

int mystery( int a, int b )
{
    if ( b == 1 ) // base case
        return a;
    else // recursion step
        return a + mystery( a, b - 1 );
} // end function mystery
```

THE END

ALL THE BEST FOR THE EXAMINATION