UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS
COLLEGE OF INFORMATICS AND ELECTRONICS

Assessment Paper

MODULE CODE: CS4013/CS4113 **MODULE TITLE**: Object Oriented Design/Programming

TERM: Autumn Repeat, 2008 **EXAM DURATION:** 2.5 hours
VALUE OF EXAM: 100%

LECTURER: Chris Exton

INSTRUCTIONS TO CANDIDATES:
Answer all questions. **(Total 100 marks)**

Part A (20 Marks) (5 Marks per question)

Please provide answers to the following question using example C++ to illustrate your answer

QA1: Explain scope in relation to public, private and protected?

QA2: How is a default constructor in a inherited call invoked by the subclass?

QA3: What is the difference between a copy constructor and an overloaded assignment operator?

QA4: What are friends in C++?

Part B (20 Marks)

You compile and execute the following program. What is the EXACT output?

QB1: (10 Marks)

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

```
class X {};

void f() {
        cout << "before throw" << endl;
        throw X();
        cout << "after throw" << endl;
}

int main()
{
        cout << "main start" << endl;
        try {
                cout << "before call to f" << endl;
                f();
                cout << "after call to f" << endl;
        }
        catch (X) {
                cout << "X caught" << endl;
        }
        cout << "main end" << endl;
}
```

QB2: (10 Marks)

You compile and execute the following program. What is the EXACT output?

```
public class Exam1
{
  public int tempInt = 65;
  public static int tempInt2 = 65;

  public Exam1()
  {
    int tempInt2 = 68;
    tempInt2++;
    System.out.println("Good morning HAL here");
  }

  public Exam1(int inValue)
  {
    super();
    tempInt2++;
    System.out.println("Welcomes");
    tempInt = 78;
    System.out.println("or is it " + ++tempInt + "?");
```

```
  }

  public static void main(String args[])
  {
    tempInt2++;
    System.out.println("Start Here");
    Exam1 temp = new Exam1(93);
    temp.start(55);
    tempInt2++;
    System.out.println("Good bye " + tempInt2);
  }

  public void start(float inValue)
  {
    tempInt2++;
    System.out.println("maybe " + ++tempInt2 + " welcomes");
  }

  public void start(long inValue)
  {
    tempInt2++;
    System.out.println("In fact " + ++tempInt + " welcomes");
  }
}
```

Part C (40 Marks)

QC1: (20 Marks)

Java V's C++ and Parameter passing

For passing parameters to functions, C++ supports both pass-by-reference and pass-by-value. In Java, parameters are always passed by value. However, in Java all non-primitive values are references to objects (in C++ terms, they are (smart)-pointers). Objects are not values in Java and only their references can be manipulated; C++ people who are used to having objects as values may confuse this with pass-by-reference.

Discuss the following in a similar manner.

Java V's C++ and Overloaded operators
Java V's C++ and Inheritance
Java V's C++ and Compilation units
Java V's C++ and the JVM
Java V's C++ and Pointers

QC2: (20 Marks)

Programming languages C and C++ are most commonly associated with memory overflows, because they provide no built-in protection against accessing or overwriting data in any part of memory and do not check that data written to an array (the built-in buffer type) is within the boundaries of that array.

Discuss how this might occur in relation to the heap and stack. Make sure to include and describe the sort of code and operations that would result in a memory overflow and their consequent effect on the heap and stack as part of your answer.