

Ollscoil na hÉireann, Gaillimh
National University of Ireland, Galway

GX 1624

Semester II Examinations, 2002/2003

Exam Code(s)	1SD1
	1MF1
	4BP1
Exam(s)	HIGHER DIPLOMA in APPLIED SCIENCE (SOFTWARE DESIGN & DEVELOPMENT)
	M.Sc. in SOFTWARE DESIGN & DEVELOPMENT
	B.E. in ELECTRONIC & COMPUTER ENGINEERING
	CT862
Module Code(s)	CT520
	CT470
Module(s)	GUI AND OBJECT ORIENTED PROGRAMMING
	OBJECT ORIENTED PROGRAMMING
Paper No.	1
Repeat Paper	Special Paper
External Examiner(s)	Prof. D. Bell
Internal Examiner(s)	Prof. G. Lyons
	Dr. S. Redfern
	Dr. M. Schukat

Instructions:

**Instructions for candidates presenting for
CT862 and CT520:**

Time allowed: 3 hours

Answer any 2 questions from section A

and

Answer any 2 questions from section B

All questions carry equal marks.

Please use separate answer books for each section.

Instructions for candidates presenting for CT470 only:

Time allowed: 2 hours

Answer all 3 questions from section A

Duration	2 hrs
	or
	3 hrs
No. of Answer books	2

Requirements:

Handout

MCQ

Statistical Tables

Graph Paper

Log Graph Paper

Other Material

No. of Pages

8

Department(s)

Information Technology

Section A (Object Oriented Programming)

Programs are to be written in C++

Q.1.

(i) 8 marks

Correct the class definition in the header file below, which contains 7 syntax errors and one possible runtime error. You'll get 1 mark for every correct answer; for every incorrect answer 1 mark is subtracted. The total amount of marks you can achieve is between 0 and 8, e.g. you don't get a negative overall mark.

```
// bug.h: Class definition of bug

#include <iostream.h>

class bug
{
private:
    int var1,
    int* var2;

public:
    // Constructor
    void bug(int k) : var1(k)
    {
        var2 = new int[k];
    }

    // Member function void assign(int)
    void assign(int j)
    {
        for (int i = 0; i < j; i++)
        {
            var2[i] = i;
        }
    }

    // Member function void print_size(void)
    void print_size(void)
    {
        cout << "var1 = " << var1;
    }

    // Prototype for int GetContentOfVar1(void)
    int GetContentOfVar1(void)

    // Destructor
    bug(void)
    {
        delete var1;
    }
}
```

(ii) 4 marks

What are the benefits of *inline functions* in C++? Provide a code fragment to illustrate your answer.

(iii) 8 marks

Prototype and implement a **ComplexNumber** class which has two attributes *RealPart* and *ImaginaryPart*. Both attributes are of the type *double*. Overload the minus ("-") and multiplication ("*") operator in order to implement subtraction and multiplication operations on instances of the **ComplexNumber** class.

Hints: If $(a + bi)$ and $(c + di)$ are two complex numbers, the equations for subtraction and multiplication are

$$(a + bi) - (c + di) = (a - c) + (b - d) i$$

$$(a + bi) * (c + di) = (ac - bd) + (bc + ad) i$$

a and c are attributes *RealPart*, b and d are attributes *ImaginaryPart*.

Write also an implementation for the output operation

friend ostream& operator<<(ostream& os, const ComplexNumber& f)

Q.2.

(i) 8 x 2.5 marks

Explain the meaning/purpose of the following terms:

- (a) const
- (b) operator overloading
- (c) default parameter
- (d) friend function
- (e) virtual function
- (f) namespace
- (g) this
- (h) <<

Q.3.

(i) 6 marks

Distinguish between functions using *pass-by-value* and *pass-by-reference*. What are the benefits and problems of references passed as arguments in functions?

(ii) 7 marks

What are constructors and destructors useful for?

Implement the constructor function and the destructor function for the following class:

```
class Person
{
private:
    char *name; // Pointer to char array, which will contain name
    char *prof; // Pointer to char array, which will contain
                // profession of person.
    int age;    // Age of person

public:
    // What to implement in constructor:
    // A) Allocate two char arrays for name and profession on the heap
    //    using strlen (see hints below) to determine their sizes. The
    //    attributes name and prof will point to these arrays.
    // B) Copy content of myName and myProfession into these arrays using
    //    strcpy (see hints below).
    // C) Update attribute age.
    Person(const char* myName, const char* myProfession,
           const int myAge);

    // Destructor
    ~Person(void);
}
```

Hints:

*int strlen(char *p)* returns the length of the string passed without the terminating character.

You probably need the string copy function *strcpy(char *destination, char *source)*.

(iii) 7 marks

What is the meaning of the terms “virtual class” and “class inheritance”? Explain the difference between deriving a class using the keyword **protected** and deriving a class using the keyword **public**.
