UNIVERSITY EXAMINATIONS

UNIVERSITEITSEKSAMENS

UNISA university of south africa

COS1512
RCO1512

May/June 2015

INTRODUCTION TO PROGRAMMING II

Duration . 2 Hours

75 Marks

EXAMINERS :		
FIRST	MRS HW DU PLESSIS MRS MA SCHOEMAN	MR ES MTSWENI
SECOND	PROF ID SANDERS	

Closed book examination

This paper consists of 6 pages and 6 questions.
Please make sure that you have all 6 pages with the 6 questions.

Instructions / *Instruksies*:

- Answer all the questıons
- Do all rough work ın the answer book
- The mark for each questıon is gıven ın brackets next to the questıon
- Please answer the questıons ın the correct order If you want to do a questıon later, leave enough space
- Number your answers and label your rough work clearly.
- Marks are awarded for part of an answer, so do whatever you are able to ın each questıon

GOOD LUCK!

[TURN OVER]

QUESTION 1 [5]

Assume the following declarations

```
char name[21];
char yourName[21];
char studentName[21];
```

For each of the following statements, explain why the statement is invalid. Also provide the correct statement(s) to achieve the desired effect.

1.1 `yourName[0] = '\0';` (1)

1.2
```
if (yourName == name)
        studentName = name;
```
(4)

QUESTION 2 [4]

2 1 What is the output of the following program? What does recursive function `b()` calculate? (2)

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

string b (int n)     //recursive function
{
    string s;
    if (n%2 == 0) s = "0";
    else s = "1";
    if (n < 2) return s; //basis
    return b (n/2) + s; //recursion
}

int main()
{
    string result = b (8);
    cout << "8 converted by function b() is " << result;
    return 0;
}
```

2.2 Identify the base case in the recursive function `b()` (1)

2.2 Identify the general case in the recursive function `b()`. (1)

QUESTION 3 [7]

3.1 Consider the following code fragment which is assumed to be embedded in a complete and correct C++ program

```
1.    int x, y, *p, *q;
2     p = new int;
3     *p = 42;
4     q = p;
5     x = *q;
6     p = &y;
7     y = 35;
```

Show diagrammatically the state of memory after

3.1.1 lines 1-5 have been executed. (3)

3.1.2 lines 6-7 have been executed. (2)

3.2 The program segment below either contains an error or a risky statement. Identify and explain the error or risky statements: (2)

```
float* q = new float(3.456);
delete q;
*q = 4.67;
```

QUESTION 4 [31]

Use separate compilation to define a class called `Donor` that represents a blood donor. This class contains three member variables:

- `name`, a string that holds the name of the blood donor
- `contact`, a string that holds the contact details of the donor
- `type`, a string that holds the blood type of the donor.

In addition, the class should contain the following member functions:

- A default constructor that initializes `name`, `contact` and `type` each to an empty string
- An overloaded constructor that accepts a new blood donor and sets `name`, `contact` and `type` to specified values.
- A destructor that does not perform any action
- Accessor functions `get_name()`, `get_contact()` and `get_type()` to return the values stored in an object's `name`, `contact` and `type` member variables respectively

- An overloaded equality operator== to compare two blood donors The == operator is implemented as a **friend** function with the following prototype.

  ```
  bool operator==(const Donor & donor1, const Donor & donor2)
  ```

  This function returns `true` if `donor1` and `donor2` have the same blood type and `false` if not

- An overloaded extraction operator >> (implemented as a **friend** function) so that it can be used to input values of type `Donor` from any file

- An overloaded insertion operator << (implemented as a **friend** function) that displays a donor's name, contact details and blood type

You should attempt the solutions as follows:

4.1 Create the header file `Donor.h` that contains the `Donor` class specification (8)

4 2 Create the implementation of the class `Donor` including all the **friend** functions. (13)

4.3 Demonstrate the class in an application program (`main()`) that is used to list and count all the blood donors of a specified blood type. Allow the user to enter the blood type for which a donor is required Use the overloaded constructor to initialise the `Donor` object `donors_needed` to the blood type the user specified (initialize the name and contact details for this object to empty strings).

All the registered blood donors are stored in a file `AllDonors.txt`. Use a `while` loop to input the donors from `AllDonors.txt`, use the overloaded equality operator== to compare the donors read from `AllDonors.txt` one by one with `donors_needed`, and display a list of all the donors that has the specified blood type. (10)

QUESTION 5 [15]

Consider the class specifications (interfaces) for the classes `Fiction` and `NonFiction` below. Class `Fiction` represents a book published as fiction, e g. a novel or poetry, and class `NonFiction` represents a non-fiction publication, e g. a journal or text book

```
class Fiction
{
public:
     Fiction ();
     Fiction (string titleP, string authorP, string genreP,
              char[4] publishedP),
     void set_title (string titleP);
     void set_author (string authorP);
     void set_genre (string genreP);
     void set_published (const char year[4]);
     string get_title ( ) const;
     string get_author ( ) const;
```

```
     void get_published (char[] year) const,
     string get_genre ( ) const;
     void display_Info( ) const, //display title, author, ISBN, genre
                                 //and published
private:
     string title;
     string author;
     string genre;  //e.g. short story, poetry, novel, etc
     char published[4];
}

class NonFiction
{
public:
     NonFiction ();
     NonFiction (string titleP, string authorP, string subjectP,
                 string KeywordsP[], char[4] publishedP);
     string get_title ( ) const;
     string get_author ( ) const;
     string get_subject ()const;
     void get_keywords (string KeywordsP[]);
     void get_published (char year[])const;
     void display_Info( ) const; //display title, author, ISBN, subject,
                                   //keywords and published
private:
     string title;
     string author;
     string subject; //e.g. computer science, maths etc
     string keywords[10];
     char published[4];
}
```

5 1 Create a base class `Book` from which both class `Fiction` and class `NonFiction` can be derived. NB: Provide only the interface for the base class. (6)

5.2 Code the interface for class `Fiction` as derived from class `Book` Override the member function `display_Info()` for class `Fiction` Do not provide any implementation. (6)

5 3 Class `Fiction` should override `display_Info( )` Is this function overloaded or redefined? Explain by referring to the difference between overloading a function and redefining a function (3)

QUESTION 6 [13]

Many application programs use a data structure called a dictionary in which one can use a key value to retrieve its associated data value. For example, we might want to associate automobile part numbers with the names of the corresponding parts

Key	Value
100000	tire
100001	wheel
100002	distributor
100003	air filter

[TURN OVER]

The following class interface presents an approach to implementing the above scenario

```
class Dictionary {
   public·
         Dictionary();
   void Add(int key, const string &value),
   string Find (int key)  const;
         private
   vector<int>  Keys,
   vector<string> Values;
};
```

The class `Dictionary` has the following operations (member functions)

- `Add()` - adds a new key and value to the dictionary
- `Find()` - retrieves the corresponding value for that particular key, for example `Find(100002)` would return "`distributor`".

6.1 Provide an interface of the template class `Dictionary<Tkey,TValue>` In other words, re-design the *`Dictionary` interface* so that it may be used to create a `Dictionary` containing keys and values of any type For instance the value could be of type `double`, whereas the key could be of type `char` Note the key and value may be most likely of different types hence we need two different template arguments to be supplied (5)

6.2 Implement the `Find()` function of the template class `Dictionary<Tkey,TValue>` Assume the key is valid and exists within the dictionary. (6)

6 3 Provide a declaration for a `Dictionary` that has keys of type `string` and values of type `double` (2)