UNIVERSITY EXAMINATIONS

UNIVERSITEITSEKSAMENS

UNISA university of south africa

CSCP1DB

January/February 2017

C++ AS SECOND PROGRAMMING LANGUAGE

Duration 2 Hours

75 Marks

EXAMINERS .

FIRST	MRS P LE ROUX
SECOND	MR CL PILKINGTON

Closed book examınatıon

Thıs examınatıon question paper remaıns the property of the Unıversıty of South Afrıca and may not be *removed from the examınation venue*

This paper consists of 7 pages

INSTRUCTIONS:

1 Answer all the questıons ın the answer book

2 All rough work must be done ın the answer book

3 Please answer the questıons ın order If you want to answer a questıon later, leave enough space

GOOD LUCK!

QUESTION 1 [6 marks]

1.1 Describe the output from the following program. (2)

```
for(int i = 5; i > 0; i--)
{
    for(int j = 0; j < i; j++)
        cout << '*';
    cout << endl;
}
```

1.2 Convert the following for loop into a while loop. (2)

```
for (int i = 1; i <= n; i++)
     cout << i * i;
```

1.3 The following code is supposed to write out the positive even numbers less than 12. That is, it will output the numbers 2,4,6,8, and 10. However, there is a problem. Explain the problem and correct the code. (2)

```
int x = 1;
while (x != 12)
{
  cout << x << endl;
  x = x + 2;
}
```

QUESTION 2 [12 marks]

2.1 In the following program explain the difference between the functions `doubleNum1` and `doubleNum2` and show how each are being called. (4)

```
void doubleNum1 (int &value)
{
   value *=2;
}

int doubleNum2 (int value)
{
   return (value *= 2);
}
```

2.2 Write a function that return the cube of the integer passed to it. For example `cube(2)` will return `8` and `cube(3)` will return `9`. Also complete the `main()` program that calls the cube function. The `main` function read in a value with which the `cube` function must be called. (4)

```
// cube function that return the cube of the given integer
// YOUR cube FUNCTION CODE SHOULD COME HERE


// main program that tests the cube function
int main()
{
       int n;
       int answer;
       cin >> n;

       // CALL cube. WRITE ONLY ONE STATEMENT

       cout << answer;
       return 0;
}
```

2.3 Write and test the function

```
int frequency (int [], int, int)
```

This function counts the number of times the item x appears among the first n elements of the array a and return that count as the frequency of x in a. Initialize the array a with 10 random integers. (4)

For example: if array a is initialized with the values 10, 30, 10, 50, 40, 90, 80, 30, 40 and 10, item is 10 and n is 8, the output will be as follows:

```
The frequency of item 10 among the first 8 elements of the array is 2
```

QUESTION 3 [4 marks]

The program below calculates pocket money for children based on their ages. Complete and correct the program by answering the questions that follow below the program.

```
1   #include <iostream>
2    using namespace std;
3    const int NUMBER = 6;
4    int main()
5    {
6      int *ptr1, *ptr2, total;
7      int factor = 20;
8      int save_age[NUMBER];
9.
10.    total = &ptr1;
11     for (int i = 0; i < NUMBER; i++)
12     {
13        cout << "Please enter the child's age  " << endl;
14        cin >> save_age[i];
15     }
16     for (int i = 0; i < NUMBER; i++)
17     {
18        ptr1 = save_age[i] * *ptr2;
19        cout << "For age  "<< save_age[i];
20        cout <<" the pocket money is " << *ptr1 << endl,
21     }
22.
23   }
```

3 1 Complete line 9 by assigning the address of `factor` to `int` pointer `ptr2` (1)

3 2 In line 10 we want to assign the address of `total` to `ptr1` Correct the error (1)

3 3 In line 18 we want to change the value of the variable to which `ptr1` is pointing to, to `save_age[i]` multiplied by the value of the variable to which `ptr2` is pointing Is this statement correct? If not, correct the error (1)

3.4 Complete line 22 by giving a C++ statement to release the memory occupied by `ptr1` and `ptr2` (1)

QUESTION 4 [4 marks]

4 1 Vectors are part of a standard C++ library known as the STL (Standard Template Library) Explain when and how you will use a vector (2)

4 2 Is the following program legal? If so, what is the output? (2)

```
#include <iostream>
#include <vector>
using namespace std,

int main( )
{
    vector<int> v(10),
    int i,

    for (i = 0, i < v size( ), i++)
        v[i] << " ",

    vector<int> copy,
    copy = v,
        v[0] = 42,

    for (i = 0, i < copy size( ), i++)
        cout << copy[i] << " ",
    cout << endl,

    return 0,
}
```

QUESTION 5 [30 marks]

Define a class `Donor` that represents a blood donor This class has 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 have 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`
- 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

5 1 Create the header file `Donor.h` that contains the `Donor` class specification (8)

5 2 Create the implementation of the class `Donor` including all the friend functions (12)

5 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 the donors should be listed 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 **read** 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 print a list of all the donors that has the specified blood type Also determine and print the total number of donors with the specified blood type (10)

[TURN OVER]

QUESTION 6 [19 marks]

The class `Competitor` below describes a competitor taking part in the Eisteddfod. Consider the class specification (interface) for the class:

```
class Competitor
{
public:
     Competitor();
     Competitor(string new_name, string new_ID, string new_item);
                          //marks and final_mark are initialized to 0
    void set_name (string new_name);
    void set_competitor_ID (string new_ID);
    void set_item (string new_item);
    void set_marks(int m[5]);
    string get_name ( ) const;
    string get_competitor_ID ( ) const;
    string get_item ( )const;
    void get_marks(int m[5])const;
    int get_final_mark ( ) const;
    void calc_final_mark( ) const;//determine final mark for competitor
 private:
    string name;
    string competitor_ID;
    string item;
    int marks[5];       //marks allocated by five judges
    int final_mark;//this is a weighted average with 50% from the 1st
                   //judge and the average of the other judges for the
                   //remaining  50%
};
```

6.1 Derive a class `MusicCompetitor` from class `Competitor`. This class has an additional member variable `instrument` that holds the instrument the competitor is playing, and additional member functions `set_instrument()` and `get_instrument`. The class should override member function `calc_final_mark()` to determine the final mark for the competitor. Only provide the class interface. (7)

6.2 Implement the overloaded constructor for the class `MusicCompetitor` by invoking the base class constructor. (3)

6.3 Consider the following implementation of the overridden `calc_final_mark()` for class `MusicCompetitor`. The final mark for `MusicCompetitors` is calculated as the average of the marks allocated by the five judges:

```
MusicCompetitor::calc_final_mark()
{
      int total =0;
      for (int i = 0; i < 5; i++)
           total += marks[i];
      final_mark = int (total/5);
}
```

When compiled, this implementation produces these two errors

```
In member function 'void MusiCompetitor  calc_final_mark()'
Error 'int Competitor  final_mark' is private
Error 'int Competitor  marks[5]' is private
```

6 3 1 Explain the reason(s) for these two errors (2)

6 3 2 Explain two different ways in which to correct these errors, and show the corresponding code fragments NB If you adapt the class Competitor **do not** copy the complete class to show how you adapt it, use the line numbers and indicate changes next to it (6)

6 4 Consider the following instantiation

```
MusicCompetitor michaelJ,
```

Write down a statement to invoke the version of `calc_final_mark()` provided in `Competitor`, for `michaelJ` (1)