

COS1512 ASSIGNMENT 4 2022

TUTORIAL MATTER:	Chapters 14, 15 and 17 of the Study Guide (Tutorial Letter 102 available under Additional Resources on the COS1512 website on myUnisa) Chapters 14 (excluding section 14.3), 15 (excluding sections 15.2 and 15.3) and 17 of Savitch
WEIGHT:	25
MARKS:	75
DUE DATE:	19 September 2022
CUT-OFF DATE	26 September 2022

NB: This assignment consists of two parts:

- a part where you write and implement program code (this part) and
- an MCQ part where you answer questions on the code you have written, and the material covered in this assignment.

The MCQ part of the assignment will be available in the Assessment Shell for Assignment 4 on the myModules site for COS1512.

You will not be able to do the MCQ part unless you have completed the coding part.

Question 1

A recursive function is a function that calls itself until a certain condition is met. The program below contains a recursive function `count()`. Run the program and answer the questions below:

```
1. #include <iostream>
2. using namespace std;
3. void count(int counter)
4. {
5.
6.     if (counter == 0)
7.         return;
8.     else
9.     {
10.         cout << "Calling counter(" <<counter << ")" << endl;
11.         count(--counter);
```

```

12.     cout << "counter(" << counter << ") processed  "
        << endl;
13.     return;
14. }
15. }
16. int main()
17. {
18.     int i = 3;
19.     count(i);
20.     return 0;
21. }

```

- (a) What is the output displayed on the console when the main program is executed?
- (b) What is the purpose of the base case?
- (c) What is the purpose of the general case?
- (d) Identify the base case in the recursive function `count()`.
- (e) Identify the general case in the recursive function `count()`.
- (f) Modify the recursive function `count()` so that it displays a countdown to 0 from the parameter sent to the function. I.e. modify the function `count()` so that for the call

```

18. int i = 3;
19. count(i);

```

in the main program, it will display the following:

```

3
2
1
0
Process returned 0 (0x0)   execution time : 1.220 s
Press any key to continue.

```

Note that the function `count()` should remain a recursive function. Do not use iteration (a `for` loop) to display the countdown from 0.

Question 2

Examine the code fragment below and answer the questions that follow:

```

1: #include <iostream>
2: using namespace std;
3:
4: //-----
5:
6: class A

```

```

7: {
8:     private:
9:         int x;
10:    protected:
11:        int getX();
12:    public:
13:        void setX();
14: };
15:
16: int A::getX()
17: {
18:     return x;
19: }
20:
21: void A::setX()
22: {
23:     x=10;
24: }
25:
26: //-----
27: class B
28: {
29:     private:
30:         int y;
31:    protected:
32:        A objA;
33:        int getY();
34:    public:
35:        void setY();
36: };
37:
38:
39: void B::setY()
40: {
41:     y=24;
42:     int a = objA.getX();
43: }
44:
45: //-----
46:
47: class C: public A
48: {
49:    protected:
50:        int z;
51:    public:
52:        int getZ();
53:        void setZ();
54: };
55:
56: int C::getZ()
57: {
58:     return z;

```

```

59: }
60:
61: void C::setZ()
62: {
63:     z=65;
64: }

```

Answer the following questions based on the code fragment given above:

- (a) Is line 18 a valid access? Justify your answer.
- (b) Is line 32 a valid statement? Justify your answer.
- (c) Identify another invalid access statement in the code.
- (d) Class C has `public` inheritance with the class A. Identify and list class C's `private`, `protected` and `public` member variables resulting from the inheritance.
- (e) If class C had `protected` inheritance with the class A, identify and list class C's `private`, `protected` and `public` members variables resulting from the inheritance.

Question 3

Consider the class definition below and answer the questions that follow:

Note: The class definition is also called the class specification or the interface for the class.

```

class Marks
{
public:
    Marks();
    Marks(string name, string number, int asg1, int asg2,
          int asg3, double test);
    double calcMark() const;
private:
    string stdtName;
    string stdNumber;
    int[3] assignments;
    double testMark;
}

```

- (a) Implement the class `Marks` and test it in a driver program.

- (b) Derive a class `FinalMark` from class `Marks`. This class has an additional member variable `examMark` that holds the examination mark. The class should override member function `calcMark()` to calculate a final mark that includes the three assignments, the test mark and the examination mark.
- (c) Implement the overloaded constructor for the class `FinalMark` by invoking the base class constructor.
- (d) Implement the member function `calcMark()` for the derived class `FinalMark`. The test mark contributes 20% to the final mark, the average of the three assignments contributes 10% and the examination mark contributes 70% to the final mark. Note that function `calcMark()` must return an integer result.
- (e) Test your class `FinalMark` in a driver program that include the following instantiation:

```
FinalMark myMark;
```

Include a statement to invoke the version of `calcMark()` provided in class `Marks` for object `myMark` and display the name and student number with the mark.

Use another statement to invoke the version of `calcMark()` defined in the derived class `FinalMark` and again display the name and student number with the mark.

- (f) Indicate whether the following statement is valid or invalid. Give a reason for your answer:

If `testmark` was a protected member variable of `Marks`, the following statements embedded in a complete main program would be legal:

```
Marks PetersMarks;

cin >> PetersMarks.testMark;
```

Question 4

- (a) Write a function called `count()` to determine the number of times a specific value occurs in a vector of integers. The function should receive two value parameters: the vector to be searched (`v`) and the value to search for (`val`). The function `count()` should return the number of times `val` occurs in vector `v`.

Test your function `count` in a program by declaring a vector and initializing it, and then call function `count` to determine how many times a specific value occurs in the vector.
- (b) Write a template version of the function `count` to determine the number of times a specific value occurs in a vector of any base type and test it by declaring two

or more vectors with different base types and determining how many times a specific value occurs in these two vectors.

Question 5

[19]

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

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”.

Consider the following implementation of the class `Dictionary` and convert it into a template class. 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.

Also test your template class by declaring two objects of template class `Dictionary` with different template arguments.

`Dictionary.h`

```
#ifndef DICTIONARY_H
#define DICTIONARY_H
#include <vector>
#include <string>
#include <iostream>
```

```

using namespace std;

class Dictionary
{
    public:
        Dictionary();
        void add(int key, const string &value);
        string find (int key)  const;
        void display();
    private:
        vector<int> keys;
        vector<string> values;
};
#endif // DICTIONARY_H

```

Dictionary.cpp

```

#include "Dictionary.h"
#include <vector>
#include <iostream>

using namespace std;

Dictionary::Dictionary()
{
    //nothing to do, vector member variables are empty on
    declaration
};
void Dictionary::add(int key, const string &value)
{
    keys.push_back(key);
    values.push_back(value);
}

string Dictionary::find (int key)  const
{
    string value = " ";
    for (unsigned int i = 0; i < keys.size(); i++)
        if (key == keys[i])
            value = values[i];
    if (value == " ")
        return "no such key can be found";
    else return value;
}

void Dictionary::display()
{
    for (unsigned int i = 0; i < keys.size(); i++)
        cout << keys[i] << ' ' << values[i] << endl;
    return;
}

```

Main.cpp

```
#include <iostream>
#include <cstdlib>
#include "Dictionary.h"
#include <vector>

using namespace std;

int main()
{
    Dictionary parts;
    string part;
    int key;

    //add 4 values to the parts dictionary
    for (int i = 0; i <= 3; i++)
    {
        cout << "Please enter a part name and a key to add to
the parts dictionary." << endl;
        cout << "Part name: ";
        getline(cin, part);
        cout << "Key for part name: ";
        cin >> key;
        parts.add(key, part);
        cin.get();
    }
    cout << endl;

    parts.display();
    cout << endl;

    //find the part for a key
    cout << "For which key do you want to find the part? ";
    cin >> key;
    cout << "The part for key " << key << " is ";
    cout << parts.find(key) << endl;

    // cout << parts.find(100002);

    return 0;
}
```

Question 6

Earlier in the year we talked about 21st century skills. One of the ways to develop critical thinking (an important 21st century skill) is to reflect on one's learning

experiences. See the quote from the Open University (<https://www.open.edu/openlearn>) below:

“When we reflect, we consider deeply something that we might not otherwise have given much thought to. This helps us to learn. Reflection is concerned with consciously looking at and thinking about our experiences, actions, feelings, and responses, and then interpreting or analyzing them in order to learn from them (Atkins and Murphy, 1994; Boud et al., 1994). Typically we do this by asking ourselves questions about what we did, how we did it, and what we learnt from doing it.”

(<https://www.open.edu/openlearn/ocw/mod/oucontent/olink.php?id=13840&targetdoc=Activity+11%3A+What+is+reflection%3F+Doc>)

This question requires you to reflect on your learning experiences while doing this assignment.

The purpose of reflection on your learning is to answer the question “What would I change to make my work better?”

Please follow this link to go directly to the questionnaire used for reflection:

<https://forms.office.com/r/3B3kMP8hn2>

“Without ambition one starts nothing. Without work one finishes nothing. The prize will not be sent to you. You have to win it.” Socrates

©

Unisa 2022