National University of Computer and Emerging Sciences

School of Computing **Hobbitesses at Shire** **For Saving the World**

CS103
Computer Programming

Course Instructor(s)

Gandalf, Frodo, Sam, Gollum, Merry and Fairy

Serial No:

Sample Final Exam

Total Time: 3 Hours

Total Marks: 96

____

Signature of Invigilator

_________________________ ________ ______ ____________________

Student Name Roll No Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have (**22**) different printed pages including this title page. There are total of (**6**) questions.
5. Use of calculator is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
7. Use **proper indentation** while writing code and make sure that your code is legible. Failing to do so can cost you marks.
8. Please read the question thoroughly and use your time **properly**, an uneven distribution of time can lead to incomplete answers.

	I	II	III	IV	V	VI	**Total**
Total Marks	36	10	15	10	15	10	96
Marks Obtained							

Vetted By: ______________________ Vetter Signature: ______________________

Question I . . . . (36 Marks)

(a) **(2 Marks)** What is the output of the following program segment? Identify errors (if any).

```
int num1 = 392;
int& num2 = num1;
num2 = 444;
num1 = num1 + num2;
int *pointer1 = &num1;
cout<< *pointer1<<endl;
```

(b) **(1 Mark)** True/False: You must declare all private members of a class before the public members.

Solution: False

(c) **(1 Mark)** Assume that *InventoryItem* is the name of a class, and the class has a *void* member function named *setPrice* which accepts a *double* argument. If *book* is an instance of the *InventoryItem* class, which of the following statements properly uses the *book* object to call the *setPrice* member function?

○ *InventoryItem::setPrice(1.49)*
√ ***book.setPrice(1.49)***
○ *book::setPrice(1.49)*
○ *book:setPrice(1.49)*

(d) **(1 Mark)** Declare an array of three *InventoryItem* objects.

Solution:

```
InventoryItem arr[3];
```

(e) **(3 Marks)** Write C++ statement(s) to dynamically allocate space for a 5-by-6 *matrix* of integers. Provide necessary declarations.

Solution:

```
int **matrix = new int *[5];
for(int i=0; i<5; i++ )
  matrix[i] = new int[6];
```

(f) **(2 Marks)** Write C++ statement(s) to free the dynamic memory allocated in the previous part.

Solution:

```
  for(int i=0; i<5; i++ )
    delete[] matrix[i];
    delete[] matrix;
```

(g) **(10 Marks)** What will the following program display on screen. Explain the error or bug if there is any.

```
#include<iostream>
#include<vector>
using namespace std;
int muffin(vector<int>::iterator i, vector<int>::iterator e, int t) {
        if (i == e)
                return 0;
        return ((*i == t) + muffin(++i, e, t));
}
int main() {
        int arr[] = { 45, 33, 45, 32, 31, 33, 45 };
        vector<int> v;
        for (int i = 0; i < 7; i++)
                v.push_back(arr[i]);
        for (unsigned int i = 0; i < v.size(); i++)
                cout << muffin(v.begin(), v.end(), v[i]) << endl;
        return 0;
}
```

Solution:

```
    3
    2
    3
    1
    1
    2
    3
```

(h) **(3 Marks)** Modify the code given below such that the following statement `o.setX(10).setY(20)` can be executed without logical and syntax error:

```
class Horizon {
public:
        void setX(int _x) {
                x = _x;
        }
        void setY(int _y) {
                y = _y;
        }
private:
        int x;
        int y;
};
int main() {
        Horizon o;
        o.setX(10).setY(20);
}
```

Solution:

```
Horizon & setX(int _x) {/*Alias is necessary*/
               x = _x;
               return *this;
       }
```

(i) **(3 Marks)** What will the following program display on screen. Explain the error or bug if there is any.

```
#include<iostream>
using namespace std;
class Maze {
private:
        int i;
public:
        Maze(int i) {
                this->i = i;
                cout << " C" << i << " \n";
        }
        ~Maze() {
                cout << " D" << i << " \n";
        }
};
Maze a(1);
int build() {
        Maze d(4);
        static Maze e(5);
}
int main() {
        Maze b(2);
        static Maze c(3);
        build();
        Maze f(6);
        return 0;
}
```

Solution:

```
C1
C2
C3
C4
C5
D4
C6
D6
D2
D5  // D5 and D3 order can be reversed.
D3
D1
```

(j) **(5 Marks)** What will the following program display on screen. Explain the error or bug if there is any.

```
#include<iostream>
using namespace std;
class Book {
public:
        void info() {
                cout << endl << ("This is a simple book ");
        }
        void info(Book &d) {
                this->info();
                d.info();
        }
};

class FunBook: public Book {
public:
        void info() {
                cout << endl << ("This is a FunBook ");
        }
};
class StoryBook {
public:
        void info() {
                cout << endl << ("This is a StoryBook ");
        }
};

class NovelBook: public StoryBook {
public:
        void info() {
                StoryBook::info();
                cout << endl << ("This is a NovelBook");
        }
};
int main() {
        Book *b = new Book;
        FunBook *fb = new FunBook;
        b->info(*fb);
        StoryBook *sb = new NovelBook;
        sb->info();
}
```

Solution:

```
This is a simple book
This is a simple book
This is a StoryBook
```

(k) **(5 Marks)** What will the following program display on screen. Explain the error or bug if there is any.

```cpp
#include<iostream>
using namespace std;
class Book {
public:
        virtual void info() {
                cout << endl << ("This is a simple book ");
        }
        virtual void info(Book &d) {
                this->info();
                d.info();
        }
};

class FunBook: public Book {
public:
        virtual void info() {
                cout<<endl<< ("This is a FunBook ");
        }
};
class StoryBook {
public:
        virtual void info() {
                cout<<endl<< ("This is a StoryBook");
        }
};
class NovelBook: public StoryBook {
public:
        virtual void info() {
                StoryBook::info();
                cout<<endl<< ("This is a NovelBook");
        }
};
int main() {
        Book *b = new Book;
        FunBook *fb = new FunBook;
        b->info(*fb);
        StoryBook *sb = new NovelBook;
        sb->info();
}
```

Solution:

```
This is a simple book
This is a FunBook
This is a StoryBook
This is a NovelBook
```

Question II . . . . (10 Marks)

(a) **(10 Marks)** Write a recursive function *pattern* that **only** receives a positive *integer* (non-zero) as argument and prints a pattern on screen. Look at the following examples to make a sense of what is required.

```
INPUT: 5
OUTPUT:
*****
****
***
**
*
```

```
INPUT: 4
OUTPUT:
****
***
**
*
```

Solution:

```
  void pattern(int n){
        if( n > 0 ){
                for( int i=0; i<n; i++ )
                        cout << "*";
                cout << endl;
                pattern(n-1);
                for( int i=0; i<n; i++ )
                        cout << "*";
                cout << endl;
        }
}
```

Question III . . . . **(15 Marks)**

(a) **(7 Marks)** Write a program that reads a file containing integers and uses *map* to display the distinct (unique) numbers in the file, named *input.txt*, with their frequencies.

For example, you can consider *input.txt* contains the following: 23 24 12 4 54 12 3 54 23

The output should be: 3:1 4:1 12:2 23:2 24:1 54:2

(b) **(8 Marks)** Write a program that reads a text file specified by the user and reports how many times each word occurred in the file. The output should be formatted in two columns, with the words in the first column and the numbers in the second column aligned on their right side in a column. You may assume the input file contains no punctuation marks. An example output is shown below for the file “in.txt”.

```
Input: In.txt

Gandalf is old
Gandalf is a wizard
Gandalf is a friend to hobbits
blah blah blah blah blah blah blah blah blah blah

Output
Enter file name: in.txt
Gandalf 3
     is 3
    old 1
      a 2
 wizard 1
 friend 1
     to 1
hobbits 1
   blah 10
```

Solution:

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

int main(){

        map<int, int> numcount;
        ifstream ifile("input.txt");
        int num;
        ifile >> num;
        while(!ifile.eof()){
                numcount[num]++;
                ifile >> num;
        }

        for( map<int,int>::iterator i=numcount.begin();
            i!=numcount.end();
            ++i)
                cout << (*i).first << ":" << (*i).second << "  ";

        return 0;
}
```

Question IV . . . . (10 Marks)

(a) **(10 Marks)** Create a class *RationalNumber*(fractions) with the following capabilities:

1. Create a constructor that two positive integers as numerator and denominator of a fraction.
2. Overload the addition, subtraction, multiplication and division operators for this class.
3. Overload the relational and equality operators for this class.

Question V . . . . (15 Marks)

Package-delivery companies, such as TCS, offer a number of different shipping options, each with specific costs associated. Your goal is to build a system for managing the package-delivery services. Your TCS company provide two types of options `TwoDayPackage` and `OvernightPackage`. Each package has following associated information of its sender and recipient, name, address, city, province and postal code. In addition each package has its associated weight(in grams) and cost per gram, this information is used to calculate the shipping cost of the package. Draw a Simley on front page to get three bonus marks.

In `TwoDayPackage` service there is additional flat fee that the shipping company charges for two-day-delivery service. Thus the total shipping cost of `TwoDayPackage` should be calculated by adding the flat fee to the weight-based cost discussed above. In `OvernightPackage` your company charges an additional fee per gram for overnight-delivery service. Thus in order to calculate shipping cost it adds the additional fee per gram to the standard cost per gram before calculating the shipping cost.

(a) **(3 Marks)** Build a hierarchical diagram of your system by identifying the main classes and the relationships among them.

(b) **(9 Marks)** Write the complete code for all the classes (except main) of your TCS system.

(c) **(3 Marks)** Given the classes of your system, now create **a function** that displays the address information of sender and receiver and calculates the shipping costs for several different Packages. The function could at most receive an array of pointers and size of the array as argument and should display the information polymorphically. It must keep track of total cost and should return the total cost of all the packages.

Question VI . . . . (10 Marks)

Given the following class and main function, write the code for storing and reading the given objects from a binary file. [Note that string class `c_str()` function returns a pointer to an array that contains a null-terminated sequence of characters.]

```
#include<iostream>
#include<vector>
#include<string>
using namespace std;
class Student {
        friend istream &operator >>(istream &, Student &);
        friend ostream &operator <<(ostream &, const Student &);
public:
        Student() {
                sid = -1;
                nsubjects = -1;
                marks = NULL;
        }
        Student(string sname, int sid, int nsubjects) {
                this->sname = sname;
                this->sid = sid;
                this->nsubjects = nsubjects;
                marks = new int[nsubjects];
        }
        ~Student() {
                if (marks)
                        delete[] marks;
        }

private:
        string sname;
        int *marks;
        int sid;
        int nsubjects;
};
istream &operator >>(istream & in, Student & std) {
        in >> std.sid;
        in >> std.nsubjects;
        in >> std.sname;
        std.marks = new int[std.nsubjects];
        for (int i = 0; i < std.nsubjects; ++i)
                in >> std.marks[i];
        return in;
}
ostream &operator <<(ostream &out, const Student &std) {
        out << std.sname << " " << std.sid << " " << std.nsubjects << endl;
        for (int i = 0; i < std.nsubjects; ++i)
                out << std.marks[i] << " ";
        return out;
}

int main() {
        Student *std;
        int x;
        cout << "Enter number of Students";
        cin >> x;
        std = new Student[x];
        for (int i = 0; i < x; ++i)
                cin >> std[i];
        // Now writes the students information to a binary file...

        // Now read the students information from that binary file
        // and display
}
```