ENSF 614 Quiz II Solutions
Fall 2021

Question 1 (6 points)

Consider following classes that uses multiple-inheritance:

```
class A{
protected:
    int a;
public:
    A(int x): a(x){}
};
class B: public A{
protected:
    int b;
public:
    B(int x, int y): A(x), b(y){}
};
class C:  public A{
protected:
    int c;
public:
    C(int x, int y): A(x), c(y){}
};
class Ball{
private:
    int ball;
public:
    Ball(int x):ball(x) {}
};
class D: public B, public C{
protected:
    int d;
    Ball round;
public:
    // constructor of class D is missing
};
```

Now answer the following questions:

Q1: If we want to use objects of class D and access some of the data members inherited from its parents, it causes an ambiguity error. In the following box explain clearly what is the reason for this issue? And, what should be done to fix this issue?

Q2: In the following space write the definition/implementation of the constructor of class D that initialises all of its data members. Don't worry about the order of argument of this constructor.

Answer Q1:

Using objects of class D cause ambiguity error as there will be duplication of data that are inherited from class A. To solve the issue, we need to make classes B and C have class A as a virtual bas class.

Answer Q2:

D::D (int a, int b, int c, int d): A(a), B(a, b), C(a, b), d(c), round(d) {}

Question 2 (9 points)

Consider the definition and partial implementation of class Point:

```
class Point{
private:
    double x, y;
    char* label;
public:
    Point(double a, double b, const char* la);
    ~Point(){delete label;}
    double get_x()const {return x;}
    double get_y()const {return y;}
    const char* get_label()const {return label;}
};
Point::Point(double a, double b, const char* la):x(a), y(b), label(new char[strlen(la)+1])
{
    assert(label != nullptr);
    strcpy(label, la);
}
void fun() {
    Point a(3, 4, "P1");
    Point b(7, 8, "P2");
    a = b;
}
int main() {
    fun();
    return 0;
}
```

This program is defective and may give a runtime error because of the last line in the function fun that is highlighted in red. In the following space write the definition/implementation of one of the missing members of class Point that causes this issue. ONLY focus on finding a fix for last line of the function fun. Also, for the simplicity purposes in this quiz, you don't need to add anything inside the definition of class Point, and you can assume all required header files are included.

Answer:

```
Point& Point::operator = (const Point& rhs) {
   if(&rhs != this) {
      delete[] label;
      label = new char[strlen(rhs.label) + 1];
      if(label == nullptr) {
          cout << "Message: memory not available..";
          exit(1);
      }
      x = rhs.x;
      y = rhs.y;
      strcpy(label, rhs.label);
   }
   return *this;
}
```

Question 3 (6 points)

In the following space write the definition of the function average, based on the following function interface comment. You may assume all required header files are included.

```
double average(const vector < vector<int>>& m);
/*REQUIRES:
 m refers to a vector of vectors that can have any number of rows, and any number of
 columns and even the number of columns in each row can be different.
 PROMISES:
 returns the average of all numbers in vector of vectors, m.
 */
```

Answer:

```
double average(const vector < vector<int> >& m) {
    double sum = 0;
    int counter =0;

    for(int i=0; i< m.size(); i++)
        for(int j =0; j < m.at(i).size(); j++){
                sum += m.at(i).at(j);
                counter++;
        }
    return sum/counter;
}
```