University of Calgary
Department of Electrical and Computer Engineering
ENSF 480: Principles of Software Design
Midterm Exam – Fall 2018 Solution

SECTION I - Multiple Choice Questions (One mark each – total 6 marks)
Please Select the Best Answer.

1. Consider the following C++ code:

```
class A{
int a;
  public:
  int bar();
};
```

```
class B{
int b;
friend class A;
public:
int fun();
};
```

```
class C {
friend class B;
int c;
public:
int foo() ;
};
```

In the above program:

   A. Class A can have access to all data members in class B and class B can have access to all data members in class C.
   B. Class A can have access to all data members in B and class B can have access to all data members in class A and C.
   C. Class B can have access to all data members in A and class C can have access to all data members in B.
   D. Class B can have access to all data members in A and class C can have access to all data members in class A and B.
   E. None of the above is correct.

2. Which one of the following group of operators must be overloaded as a member function in a C++ class? In other words they cannot be defined as a non-member global function.

   A. ==, >=, <=
   B. &&, ||, new
   C. ++, and --
   D. (), =, []
   E. All of the above
   F. None of the above.

3. A derived class inherits all the member functions of each of its base classes except:
   A. the constructors and copy constructors
   B. the destructor
   C. the assignment operator.
   D. A, B, and C are all correct answers
   E. None of the above is a correct answer.

4. An abstract class in C++ is a class that:
   A. Contains a virtual function
   B. Contains a virtual base class
   C. Is derived from a base class
   D. A, and B are correct answers
   E. None of the above is a correct answer

5. Consider the definition of the class String in C++ and assume all member functions are properly defined:

```
class String {
  public:
      String(int a = 0);
      String(const char* b);
  private:
      int length;
      char* storage;
 };
```

Which of the following statements is a **valid** statement?

   A. String s1(50);
   B. String s2 = 100;
   C. String s3;
   D. String s4 = "123";
   E. A, B, and C are all **valid** statements
   F. A, B, C, and D, are all valid statements

6. Which one of the following statements is a correct statement in C++?

   A. An overloaded operator function must be always a member of a class or a friend of a class
   B. An overloaded operator function must have at least a class objec as its argument.
   C. An overloaded operator cannot return a pointer, but can return a reference.
   D. A and B are correct answers
   E. None of the above is a correct answer

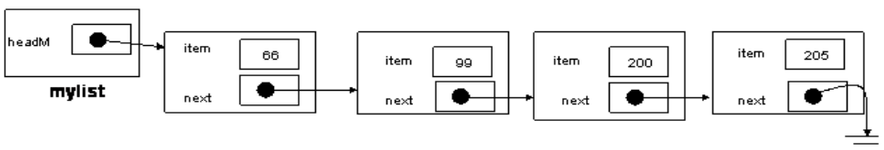
Section II – C++ Templates and Overloading Operators (11 marks)

This problem concerns a template class called *List* that maintains a list of objects of different types. Partial definition of class *List* and class *Node* in C++ are given.

```
template  <class T>
struct Node {
  T item;
  Node <T> *next;
};
```

```
template <class T>
class List {
public:
  List() {headM = nullptr;}
  void insert(T & itemA);
  friend ostream& operator << <T>(ostream& os, const List<T> x);
private:
    Node <T> *headM;
};
```

Part a (6 marks) – In the following space write the definition of `operator[]`, for class `List`. By using this operator you should be able to retrieve or modify the value of the item in the i^{th} node of the list. For example, if `mylist` is an instance of `List <int>` with four nodes as follows,



you should be able to write the following statement to retrieve the value of the first node:

```
int y  = mylist[0];
```

Or, to change the value of item in the third node from 200 to 1000, you should be able to write:

```
mylist[2] = 1000;
```

Note: If the index number is less than zero or greater than number of nodes in the list, or if the list is empty this function should give the following message and return zero: "`Something went wrong: Out of Bound Error.`"

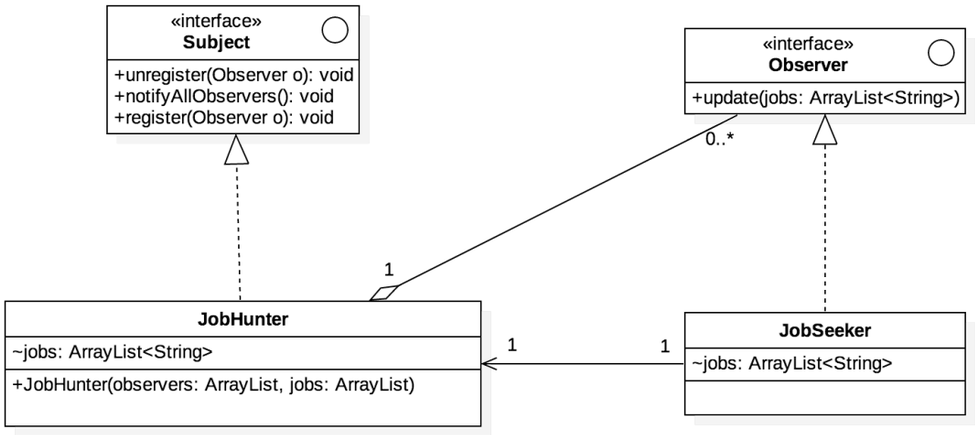
```
template <class T>
T& List<T>::operator[] (int index) {
   if(index < 0 || headM == nullptr) {
      cerr << "Something went wrong: Out of Bound Error.";
      exit(1);
   }

   int i = 0;
   Node <T>* p = headM;

   while(p != nullptr && i < index) {
      i++;
      p = p -> next;
   }

   if(p == nullptr) {
     cerr << "Something went wrong: Out of Bound Error.";
     exit(1);
   }
   return p -> item;
}
```

Part b (5 marks) – Write the definition of overloaded operator << to print entire values in the list (one value per line):

```
template <class T>
ostream& operator<< (ostrem& os, const List<T>& x) {

   Node<T>* p = headM;
   while (p)
     os << p -> item;
     p = p -> next;
   }

   return os;
}
```

SECTION III – Design Patterns (15 marks)

This question is similar to an exercise that we discussed during one of the lectures. Based on the following class diagram write the definition of class JobHunter and JobSeeker in Java. Please only implement the constructors of these classes, and their methods: `register, notifyAllObservers`, and `update`. You don't have to worry about other functions in these classes.



Note: Some of the data members of classes JobHunter and JobSeeker are shown in this diagram. You need to pay attention to relationship among the classes and consider more data members if needed.

Write the definition of your classes in the following space:

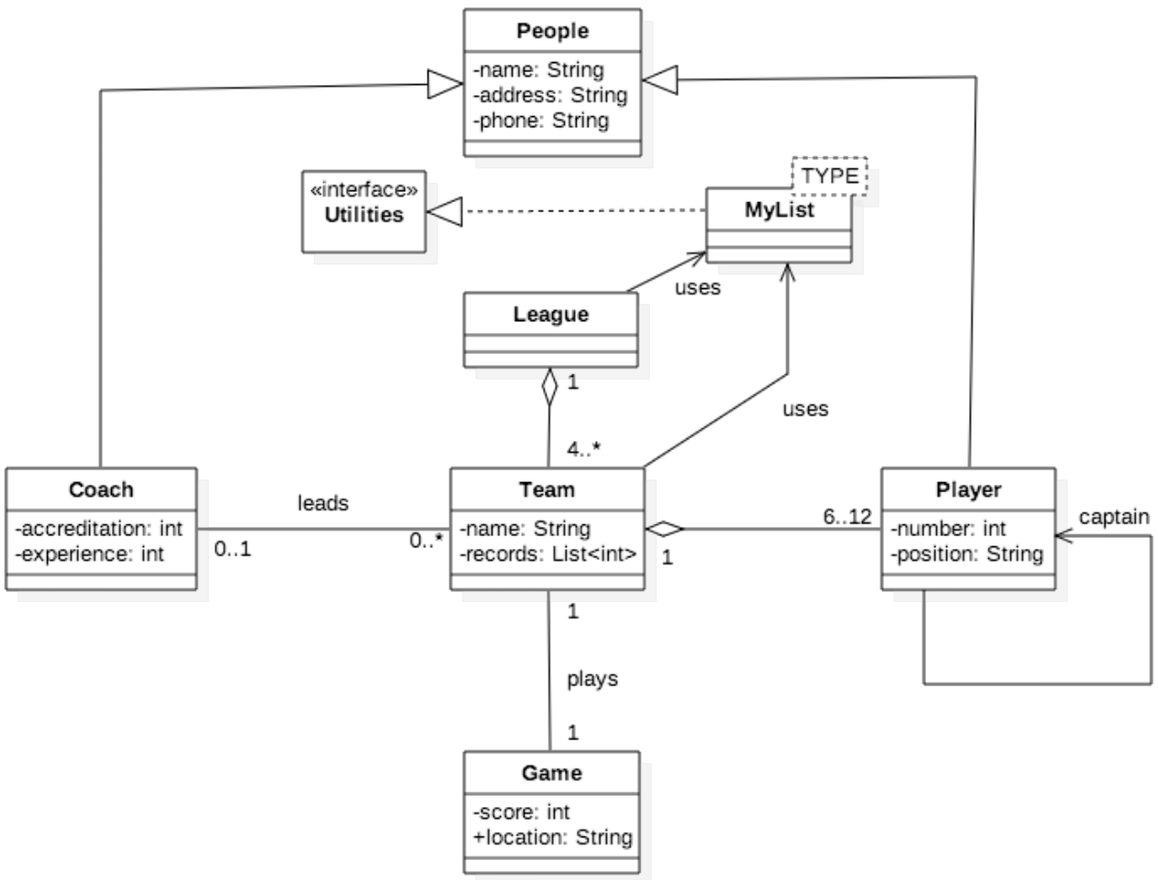
```
public class JobHunter implements Subject { // 9 marks
   ArrayList<String> jobs;
   ArrayList<Observer> observers;

   public JobHunter(ArrayList<Observer> o, ArrayList<String> j) {

       // For this exam a simple assignments of references were accepted.
       // Other possible solution is to make copies or clones of the sources (o, and j)
       jobs = j;
       observers = o;
   }

   @Override
   public void register(Observer o) {
       observers.add(o);
       o.update(jobs);

   }

   public void notifyAllObservers() {
       for(int i = 0; i < observers.size(); i++){
             Observer o = observers.get(i);
             o.update(jobs);
       }
   }
}


public class JobSeeker implements Observer {  // 6 marks
ArrayList <String> jobs;
Subject subject;

   public JobSeeker(Subject s) {
      subject = s;
      subject.register(this);
   }
   @Override
   public void update(ArrayList<String> jobs) {
      this.jobs = jobs;
   }
}
```

SECTION IV – Drawing UML Diagrams (25 marks)

Part a (15 marks) - Draw a UML class diagram (using UML 2.5 notation and syntax), for the following problem statement. For this question you don't have to show class operations, but you should show class attributes.

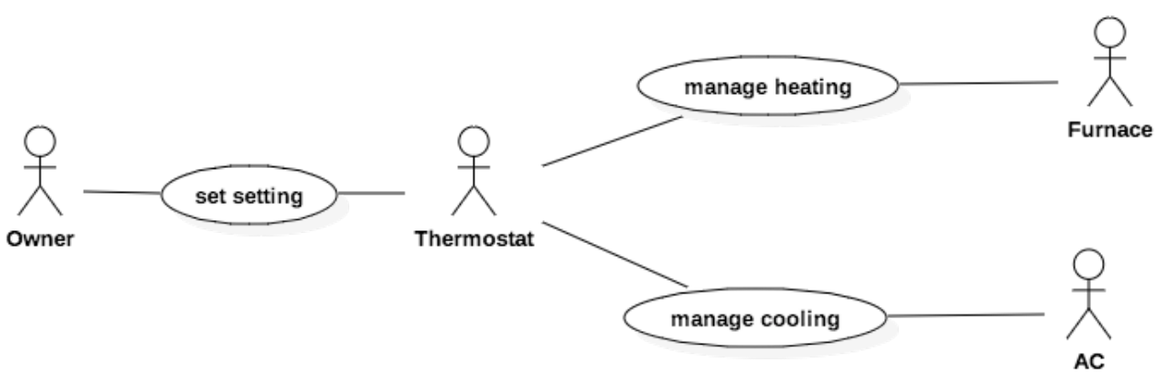
Problem Statement: A Hockey-League is made up of at least four Hockey Teams. Each Hockey-Team has six to twelve players, and one of the players assumes the role of a Captain (*Note: Captain has no additional attribute and no additional functionality – his/her attributes and functionalities are identical to other players*). Each team has a name and a list of its records (an integer `List`). Players have a number (`int` type) and a position (`String` type). Hockey teams play games against each other. Each game has a score and a location. Teams are sometimes lead by a coach. A coach has a level of accreditation and a number of years of experience (both `int` types), and can coach multiple teams. Coaches and players are classified as a person. A person, has names, address, phone number (all `String` types).

Draw a class diagram based on the given information, and be sure to label all associations with appropriate multiplicities. Apply inheritance, and aggregation (don't worry about composition), or simple association as needed. Also, in your diagram consider and show a generic class called `MyList`. Whenever a class needs to have an aggregation relationship, use `MyList` objects. Class `MyList` is supposed to realize an interface called `Utilities`.



Note: to show Captain relation a role specified on the line between player and team is also accepted.

Part b (5 marks) – In the following space draw a use-case diagram that shows the actors and use cases needed in a simple heating and cooling system that:
The owner of the house can set the temperature settings on the Thermostat.
Thermostat manages the heating device (Furnace}
Thermostat manages the cooling device (AC)



Part c (5 marks) – In the following space draw a sequence diagram between a bank-customer and the ATM-System that shows details of a use case called `withdraw-money` (withdrawing money from a bank-account). You don't have to show the bank database system in this diagram;