Final

Note

This is an open-book exam. You may consult any sources (including online ones), but discussion with others is strictly forbidden.

Problems

1. Suppose you are working on the authentication module of an online game. The authentication module contains a session Id generator. When a user logs in successfully, the server assigns a session Id that is unique among all the online users. Here is the interface of the session Id generator:

```
class SessionId{
// The details of SessionId goes here.
};

class SessionIdGenerator {
public:
   SessionIdGenerator();
   ~SessionIdGenerator();
   SessionId* GenerateSessionId();
};
```

a) (3%) The interface of SessionIdGenerator is error-prone: it does not prevent the user of the class from generating duplicate session Ids. Please describe a scenario in which the system generates duplicate session Ids.

b) (2%) What design pattern can be applied to fix the bug in the design?

c) (5%) Please provide your design to fix the bug (C++ or Java).

2. A multi-threaded system typically has some classes to make writing multi-threaded code easier. A code snippet to create a thread for some purpose looks like the following:

```
// Create and start a new thread.
AbstracThread* thread = new Thread();
thread->Start();

// The thread is now running and can start receiving tasks to perform.
// Create a concrete task and dispatch the task to the new thread.
AbstractTask* task = new MyTak();
thread->Dispatch(task);
```

The interfaces for AbstracTask and AbstractThread are as follows:

```
class AbstractTask {
public:
   virtual void Run() = 0;
};

class AbstractThread {
public:
   virtual void Start() = 0;
   virtual void Shutdown() = 0;
   virtual void Dispatch(AbstractTask* task) = 0;
   virtual void DispatchWithDelay(AbstractTask* task, int delayInMilliseconds) = 0;
};
```

And the task to perform on the thread, MyTask looks like:

```
class MyTask: public AbstractTask {
public:
   void Run() {
      // Code to perform the task on the thread.
   }
};
```

a) (4%) What design pattern is used here so that the user of AbstractThread can provide an action to be performed right away or later on the thread?

Suppose you want to use class AbstractThread and Thread to replace your existing threading library because it has higher quality than yours. Your threading classes look as follows:

```
Class AbstractRunLoop {
public:
   virtual void Start() = 0;
   virtual void Stop() = 0;
   virtual void PostTask(AbstractTask* task) = 0;
   virtual void PostDelayedTask(AbstractTask* task, int delayInMilliseconds) = 0;
};
```

b) (6%) Please design an object adapter of AbstractThread in class diagram so that you can use the implementation of AbstractThread without modifying your existing code using AbstractRunLoop. Please also show the detail of method PostTask() in your design.

3. (20%) Construct an abstract data model for a conference center management system that has to meet the following requirements:

- ✓ The conference center has several meeting rooms with different capacities.
- ✓ The conference center also has guest rooms of different types (single, double, etc.) for conference/workshop participants.

- ✓ A conference/workshop typically lasts for a few days.
- ✓ Reservations for rooms may be made several months up to one year in advance.

Please use the UML as much as possible when describing the model.

4. (10%) Cross-Site Scripting (XSS) is one of the most common types of Web application security vulnerabilities/risks.

a) What is the main cause of this security risk?

b) How can such risks be prevented?

5. (10%) Answer the following two questions regarding software development.

a) How are validation and verification related?

b) How are testing and verification related?

6. Please provide a precise description, using logical formulae, for each of the following requirements. The functions/constants and predicates you may use are: 0, 1, <, =, ≤, plus those introduced in the requirement statements. Make assumptions where you see necessary.

a) (10%) The elements from A[0] through A[i] in array A[0..N-1] (of integers, indexed by 0 through N-1) are sorted in the ascending order.

b) (5%) Array B is a permutation of Array A (assuming both store N integers, indexed by 0 through N-1).

7. Below is an implementation of the Bubble sort.

```
void bubblesort(int A[], int n)
{
   int i, j, tmp;

   j = n-1;
   // {inv: …}
   while (j>0) {
      i = 0;
      // {inv: …}
      while (i<j) {
         if (A[i]>A[i+1]) {
            tmp = A[i];
            A[i] = A[i+1];
            A[i+1] = tmp;
         }
         i++;
```

```
        }
        j--;
    }
}
```

a) (10%) Write a loop invariant (using logical formulae) for the first while loop. The loop invariant should be strong enough to deduce that, upon termination of the while loop, the array is sorted. (The functions/constants and predicates you may use are: 0, 1, $<$, $=$, $\leq$. Make assumptions where you see necessary.)

b) (5%) Write a loop invariant (using logical formulae) for the second while loop. The loop invariant should be strong enough so that, after completion of the second while loop and the operation j--, one can deduce the first loop invariant.

8. (10%) In the automata-based model checking, automata (representing systems/modules or properties) need to be composed either by asynchronous product or synchronous product. Why do we need two different kinds of composition? Please try to explain as comprehensively as possible.