Review Test Submission: Quiz 5 - Abstract Data types

User	Kaidi He
Course	201503_Advanced Software Paradigms_CSCI_6221_11
Test	Quiz 5 - Abstract Data types
Started	10/7/15 6:11 PM
Submitted	10/7/15 6:39 PM
Due Date	10/7/15 6:40 PM
Status	Completed
Attempt Score	80 out of 100 points
Time Elapsed	27 minutes out of 30 minutes
Results Displayed	All Answers, Feedback, Incorrectly Answered Questions

Question 1

15 out of 15 points

What are the two kinds of abstraction in programming languages?

Answers:
- Pseudo-code abstraction
- All answers are wrong
- Complexity Abstraction
- Process Abstraction
- Data abstraction

Response Feedback: The two fundamental kinds of abstraction in contemporary programming languages are process abstraction and data abstraction.

Question 2

10 out of 10 points

Regarding Friend classes and Friend function, indicate

what is true about the use of friendship vis-a-vis ADT.

Answers: C++ is one of the few OO programming languages that support multi-inheritance and friendship amongclasses. Although, Java does not provide multi-inheritance nor friendship among classes, these ADT concepts can be implemented by combining Java single-inheritance, Java interfaces, and private and protected modifiers.

Java does not have friend functions or friend classes, however, classes defined in the same package are implicitly friends.

Friendship among classes improves information hiding.

In C++, a sub-class is always friend of its super-classes.

Question 3

15 out of 15 points

The following is an example of a user-defined ADT in Java (see section 11.4.4.1 An Example - Sabesta's book)

```
class StackClass {
private int [] stackRef;
private int maxLen,
topIndex;
public StackClass() { // A constructor
stackRef = new int [100];
maxLen = 99;
topIndex = -1;
}
public void push(int number) {
if (topIndex == maxLen)
System.out.println("Error in push—stack is full");
else stackRef[++topIndex] = number;
}
public void pop() {
if (empty())
System.out.println("Error in pop—stack is empty");


else --topIndex;
}
public int top() {
if (empty()) {
System.out.println("Error in top—stack is empty");
return 9999;
}
else
```

```
return (stackRef[topIndex]);
}
public boolean empty() {return (topIndex == -1);}
}
```

How would be the best way to improve this ADT:

Answers:

- Divide this ADT in two files: one containing the interface -- that would include attributes and method signatures --, and another file with the realization of the interface -- that would contain the realization of the methods. This would improve encapsulation.
- You should provide accessors methods, sometimes called getters and setters, that would allow clients direct access to the so called hidden data—a better solution than simply making the data public, which would provide direct access. Getters and setters would be public. By doing so, stackRef, maxLen, and topIndex would be manipulated via these getters and setters methods by other classes.
- all answers are wrong
- Add a destructor and call the destructor to liberate memory when objects belonging to this class are not longer needed.

Response Feedback: In C++, classes can be used for encapsulation
–The class is used as the interface (prototypes)
–The member definitions are defined in a separate file

Question 4

15 out of 15 points

Regarding the support for abstract data types in programming languages, indicate what sentences are true

Answers:

- The data defined in a C++ class are called **data members**; the functions (methods) defined in a class are called **member functions**.

In order to support information hiding, a class can contain hidden and visible entities (meaning they can either hidden form or visible to clients of the class). For instance, Java provide **private**, **public**, and **protected** modifiers.

All answers are wrong

OO Pprograming Llanguages like Java provide class constructors and destructors. Destructors are often used as a debugging aid, in which case they simply display or print the values of some or all of the object's data members before those members are deallocated. The name of a destructor is the class's name, preceded by a tilde (~).

Neither constructors nor destructors have return types, and neither use return statements. In some programming languages, both constructors and destructors can be explicitly called.

Response Feedback: Access level modifiers determine whether other classes can use a particular field or invoke a particular method. There are two levels of access control:

- At the top level—public, or *package-private* (no explicit modifier).
- At the member level—public, private, protected, or *package-private* (no explicit modifier).

Question 5

0 out of 15 points

The most relevant features of Abstract Data Types (ADTs) are:

Answers: (1) All of the class instances of a class share a single copy of the member functions; (2) Each instance of a class has its own copy of the class data members; and (3) Instances can be static, stack dynamic, or heap dynamic

All answers are wrong

(1) Interface containers; (2) Implementation containers; (3) Method prototypes form

(1) Data; (2) Functions; and (3) Encapsulation.

Response Feedback: Key features of ADTs are the packaging of data with their associated operations and information hiding. The term *encapsulation* is often used interchangeably with information hiding.

Question 6

15 out of 15 points

Regarding user-defined generic classes in Java, indicate what sentences are true

Answers:
- Given the support for wildcard classes, user-defined generic classes in Java are very flexible. For instance, Collection<AbstractClass> is a wildcard class for all collection classes.
- User-defined generic classes in Java are specially usefull since they can store primitives, such as int, float.
- All answers are wrong.
- User-defined generic classes in Java are easy to be manipulated because they can be indexed.

Response Feedback: Java 5.0 supports wildcard classes. For instance, Collection<?> is a wildcard class for all collection classes. This allows a method to be written that can accept any collection type as a parameter. Because a collection can itself be generic, the Collection <?> class is in a sense a generic of a generic class.

Question 7

10 out of 15 points

Indicate what definitions of *abstract data type* are correct

Answers:
- In Java, abstract data types can be realized using classes and interfaces. Classes can be specialized by sub-classes.
- The representation of objects of the type is hidden from the program units that use the type, so the only direct operations possible on those objects are those provided in the type's definition.

The declarations of the type and the protocols of the operations on objects of the type, which provide the type's interface, are contained in a single syntactic unit. The type's interface does not depend on the representation of the objects or the implementation of the operations. Also, other program units are allowed to create variables of the defined type.

An abstract data type is a classification identifying one of various types of data, such as real, integer, bollean, that determines the possible values for that type; the operations that can be done on values of that type; the meaning of the data; and the way values of that type can be stored

Response Feedback: data type that satisfies the following conditions:

- The representation of objects of the type is hidden from the program units that use the type, so the only direct operations possible on those objects are those provided in the type's definition.
- The declarations of the type and the protocols of the operations on objects of the type, which provide the type's interface, are contained in a single syntactic unit. The type's interface does not depend on the representation of the objects or the implementation of the operations. Also, other program units are allowed to create variables of the defined type.

Wednesday, October 7, 2015 6:40:01 PM EDT

← OK