OOP[5] AND ADT[6]

[5] *Object-oriented Programming*

[6] *Abstract Data Type*

[7] *p20, Carrano and Prichard, but other correct answers are possible.*

1. [7]

   (a) Encapsulation: Objects combine data and operations
   (b) Inheritance: Classes can inherit properties from other classes
   (c) Polymorphism: Objects can determine appropriate operations at execution time.

2. An ADT means a collection of data of certain type (e.g. integers, boolean, or a combination of primary types such as student record) and a set of operations on that data.

   For example, a ADT list of integers means a collection of integers and some typical operations on the integers such as:

   - Create an empty list (of integers)
   - Determine whether a list is empty
   - Determine the number of integers on a list
   - Add an integer at a given position in the list
   - Remove the item at a given position in the list
   - Search for a specific integer in the list

   A Data Structure is a construct (within a programming language such as Java) that stores a collection of data. We have to choose a particular data structure when we implement an ADT.

   For example, we can use an array to implement an ADT list (linked list) in Java (as shown in lecture).

   By ADT list, we normally mean not only the logical relationship among the data, but also the operations on the data. By list, we may just mean the way the data stores.

3.
```
CreateAppointmentBook()
/* Creates an empty appointment book. */

isAppointment(date, time)
/* Returns true if an appointment exists for the date */
/* and time specified; otherwise returns false.       */

makeAppointment(date, time, purpose)
/* Inserts the appointment for the date, time and purpose     */
/* specified as long as it does not conflict with an existing */
/* appointment.                                               */
/* Returns true if successful, false otherwise.               */

cancelAppointment(date, time)
/* Deletes the appointment for the date and time specified. */
/* Returns true if successful, false otherwise.             */

checkAppointment(date, time)
/* Returns the purpose of the appointment at                        */
/* the given date and time, if one exits. Otherwise, returns null. */
```

   Note: other operations on the appointments are welcome.