Assignment 6
(**Due date: 10/29/2009/Friday, in class**)

| Your name: | Grade: |
|---|---|
| | |

**Important notice** on how to submit and grade this assignment:

- Write your answers on **different papers** from the question sheets; otherwise, they will **NOT** be graded.
- You do **NOT** have to write the question text, but you need to **write the question number** for each question.
- Put your solutions in the **same order** as the questions appear on the assignment; otherwise, **missed or misplaced** solutions will **NOT** be graded.
- **How to Grade:**
  - The total score for the assignment is **100** points.
  - **An extra 8%** will be added to the **TYPEWRITTEN** submissions.
  - **3 points will be deducted** from your total score if you **missed any ONE** of the following (this is a *cumulative penalty*, e.g., 9 points will be taken for 1 missed name and 2 missed required blank lines):
    - **Your name** and **assignment number** on the top of each solution sheet/paper,
    - At least **one blank line** between solutions of adjacent questions (**except for** those of *Multiple Choice* or *True/False* questions).

The following questions are taken from the textbook Chapter 6 (p. 189-193).

- **For questions 1 through 16, using A, B, C, D, E, or F as your answers** for each of these questions (you may write text solutions alongside these A, B, …F). **50%** will be deducted if your solutions are **NOT** one of these A, B, …F (even though your texts give the correct answers).
- Do **NOT** use any calculators for the conversion/calculation questions.

Exercises 1–10, match the activity with the phase of the object-oriented methodology (by **using A, B, C, or D**).

A. Brainstorming
B. Filtering
C. Scenarios
D. Responsibility algorithms

1. Reviewing a list of possible classes, looking for duplicates or missing classes
2. Asking "what if" questions
3. Assigning responsibilities to classes
4. Generating a first approximation to the list of classes in a problem
5. Assigning collaborators to a responsibility

6. Developing algorithms for the responsibilities listed on a CRC card

7. Output from this phase is a fully developed CRC card for all classes

8. Output from this phase is the OOD ready to be translated into a program.

9. During this phase, inheritance relationships are established

10. Phase in which functional programming techniques are appropriate

For Exercises 11–16, match the term with its definition (by **using A, B, C, D, E, or F**).

A. Information hiding
B. Abstraction
C. Data abstraction
D. Procedural abstraction
E. Control abstraction
F. Encapsulation

11. The bundling of data and actions so that the logical properties of data and actions are separated from their implementation

12. The practice of hiding the details of a module with the goal of controlling access to the details of the module

13. A model of a complex system that includes only the details essential to the viewer

14. The separation of the logical view of an action from its implementation

15. The separation of the logical view of a control structure from its implementation

16. The separation of the logical view data from its implementation

17. List the four steps in Polya's How to Solve It List.

19. List the problem-solving strategies discussed in this chapter.

20. Apply the problem-solving strategies discussed in the chapter to the following situations.

c. Buying a dress or suit for an awards banquet at which you are being honored

22. What is an algorithm?

23. Write an algorithm for the following tasks.

    a. Making a peanut butter-and-jelly sandwich.

24. List the three phases of the computer problem-solving model.

26. Describe the steps in the algorithm development phase.

27. Describe the steps in the implementation phase.

28. Describe the steps in the maintenance phase.

31. Describe the top-down design process.

33. Write a top-down design for the following tasks.

    b. Organizing an awards banquet for your soccer team

40. List the identifiers and tell whether they named data or actions for the designs in Exercise 33.

48. Distinguish between an object and an object class.

56. What is a programming language?

57. Distinguish between syntax and semantics.