

CS2102, B11

Exam 1

Name:

You have 50 minutes to complete the problems on the following pages. There should be sufficient space provided for your answers.

If a problem asks you to create a class hierarchy, we are looking for the interfaces, classes, and abstract classes that you would create for the problem. In particular:

- Include **implements** and **extends** statements
 - Include field names and types
 - Include method headers (names, return type, and input parameter types)
 - Full credit requires that all types and implements/extends relationships are clear. Be sure your work is clear if you use class diagrams instead of Java syntax.
 - You may omit constructors
 - You may omit method bodies unless a question asks otherwise
 - You may omit the `Examples` class (examples of data and test cases) unless a question asks otherwise
-

Grading Summary

Exam starts on the next page

Topic	Max Points	Score
Q1: Choose an appropriate data structure	20	
Q1: What is a	5	
Q1: What is a	5	
Q1: What is a	5	
Q2: What is an interface used for	7	
Q2: What is an abstract class used for	7	
Q2: When to use interface versus abstract class	6	
Q2: Create a class hierarchy	20	
Q3: Test methods with multiple correct answers	25	
Total (out of 100)		

(exam starts on the next page)

1. **Topic: Data Structures**

You are writing a simple application to remind you about appointments. Each appointment has a date and a description. Your application will alert you to your next scheduled appointment. It will **not** show you all of your appointments or help you search for available appointments. It only shows you your next appointment.

- (a) Which of the following ADTs is best suited for this application, given its intended functionality? (ignore potential extensions.) Justify your answer in a sentence or two.

Sets Bags Stacks Queues Priority Queues

- (b) Which of the following data structures would you prefer to use for this application, thinking ahead to a future version that shows all appointments and helps you search for appointments? Justify your answer in a sentence or two.

List Sorted List Binary Search Tree AVL Tree Heap

- (c) If your boss demanded that you use either a binary search tree or an AVL tree as your underlying data structure for the original application, would you have a preference between them? Explain your answer in a sentence or two.

(exam continues next page)

2. Topic: Class Hierarchies (Classes, Abstract Classes, Interfaces)

A company that sells toys needs to create a class hierarchy to manage information about their products. The class hierarchy needs to capture the following information:

- Whether a toy is electronic or made of wood. Electronic toys each need a certain number of batteries. Every toy has a recommended number of people who can use it at the same time.
 - Whether a toy is meant for children, adults, or both. Toys for children must indicate the minimum age the toy is designed for.
 - Features of each toy. Initial features are (1) whether it is for outdoor sports, and (2) whether it requires assembly (someone to put it together) before use. This set of features is expected to grow as the company develops more products.
- (a) Provide a class hierarchy (a collection of classes, abstract classes, and interfaces) for this information. Include fields and method headers as needed to capture the required information (indicate clearly which class/interface each of these should be in). Include all concepts underlined on the description above.

(exam continues next page)

- (b) Using the components of your hierarchy, show how to define a class for a tennis racket that is made of wood, is for all ages, and is for outdoor sports. Provide only as much code as is needed to show how to use your class hierarchy: in particular, you do NOT need to define any methods or additional information beyond what your class hierarchy requires in the tennis racket class.

(exam continues next page)

3. Topic: Testing functions with multiple correct answers

You are writing software to help students schedule classes. In this system, a student names four courses that they would be willing to take and the scheduler proposes a schedule containing three of those courses (including their times). A returned schedule is *valid* if each course was one of those requested and if no two courses meet at the same time.

The following code fragment shows Java classes with the essential fields and methods.

```
// The primary scheduler class.  Students call makeSched to build a schedule.
// Each option names a course that the student wants to take (like "CS2102").
class Scheduler {
    ...
    StudSched makeSched(String option1, String option2,
                       String option3, String option4) {
        ...
    }
}

// the output of makeSched is an object with three courses.
class StudSched {
    Course course1;
    Course course2;
    Course course3;
}

// A course contains a name (e.g., "CS2102") and the lecture hour as an
// integer (e.g., 11 for 11am or 13 for 1pm).
class Course {
    String name;
    int time;
}
```

Outline how you would test that `makeSched` produces valid schedules. A good answer will (a) describe any additional methods you need to write to support testing and (b) give one concrete example of a `checkExpect` that illustrates your testing approach. For the methods, provide only an input type, output type, and a description of what the method computes (don't write the method body). For the `checkExpect`, show Java expressions for the actual and expected answers, but ignore all the other surrounding syntax.

This question is not asking you to write code. Assume that the database of courses and times exists elsewhere (do not try to define it), and that you are only testing course names that exist in the database. Ignore cases in which there is no schedule using the given courses.

(space for answer on next page)

space for answer to question 3

(end of exam)