

Exam Review, Wednesday, December 7, 2011

1. Examples may include previous material: use of one-dimensional arrays, methods, control structures such as if statements and loops. You must understand the use of these.

2. Data Classes

model objects in the real world with state and behaviors
with fields and methods

3. Class definitions

a class definition is a template to create instances of the class
fields, constructor method, other methods

Example:

```
public class AssignGrade
{
    public int score;
    public String remarks;
    public AssignGrade ...
    // other methods
    public void correctGrade(int newgrade) { ... }
}
```

4. Using classes

declare a variable of that type:

```
AssignGrade agrade;
```

create an instance:

```
agrade = new AssignGrade ( 80, "do better!" );
```

(be able to explain what happens when the constructor method is called)

accessing fields in the class :

```
agrade.score, agrade.remarks
```

using methods:

```
agrade.correctGrade( 99 );
```

5. Multiple Classes

be able to use a class within another class

6. a. Explain scope rules for variables and understand examples.

6. b. Explain the uses of the public and private access keywords (don't need to do protected)

7. Be able to define a method header, or give a method definition for a method in a class

8. Declare a one-dimensional array of class instances

```
AssignGrade [ ] grades = new AssignGrade[50]
```

9. Principals of Object-Oriented Programming:

- a. Data Encapsulation – goals are modularity and data hiding
data hiding means that many fields are private and accessed through accessor methods – the get methods
- b. Inheritance – what is it? A subclass can be defined from a superclass by inheriting all of its fields and methods
- c. Polymorphism – a method can be given different definitions for subclasses as for superclasses (that is, the methods have the same name)

10. Forms of searching – give algorithms for finding the first item in an array that matches the search string, give algorithm to find all occurrences, be able to say if no items found that matches the search

11. Should be able to give algorithm to find the largest or smallest elements in an array

12. String methods used in matching:
equals(), startsWith(), toLowerCase(), indexOf(), contains
be able to describe these methods, understand examples, or use them in a search

13. Know how to swap two variables, x and y.

14. Understand bubble sorting algorithm: Suppose that we are sorting an array of integers. Given some numbers in the array, show the values in the array after each pass. Understand how the algorithm works.

15. Why are there different sorting algorithms?

16. What is an API? How do you use a programming API? Why is using an API important?

17. Be able to write a small program as specified using files and arrays with classes, for example, reading a one-dimensional array of class instances from the data in a file, counting how many are read and processing the elements of the array for some type of task.

Example: Suppose that a Scanner `sc` has already been created to read a csv file that has one int and one string on each line. Code excerpt to read the file before creating instances of the class:

```
int number; String s;
while (sc.hasNext())
{ // read all items on one line from the file here
  // note the simplified code that doesn't have if statements with hasNext(), etc.
  number = sc.nextInt();
  s = sc.next()

  // put number and s into an instance of a class and save to an array
  ...
}
```

Closed Book Exam given in iLMS – Except for Summary Sheets: two sheets, i.e. four sides of paper – prepared by yourself.