

Exercise Set 2

Prepared by: Neeraj, Julien, Mansoor
February 24, 2009

Questions from the Book

Do questions R-2.11 and C-2.1

Questions on Interfaces

1. Can we pass an interface as parameter to a method? If yes, how does this work?
2. Can an interface contain any instance variables? If yes, how do we define such variables?
3. What is the default access modifier of the methods of an interface?
4. What is the difference between an interface and an abstract class?
5. What will the code below output? Note that for concrete and abstract classes, the default access modifier for a method is "package". (Read more about access modifiers at www.cim.mcgill.ca/~langer/250/Tutorial-modifiers.pdf)
 - a) The code does not compile because class Test attempts to override a private method
 - b) The code does not compile because class SuperTest does not implement the method g()
 - c) The code compiles fine and prints out:
Sub
g
 - d) None of the above

```
interface Inter {
    public void g();
}
abstract class SuperTest implements Inter {
    private void f() {
        System.out.println("Super");
    }
}
class Test extends SuperTest {
    void g() {
        System.out.println("g");
    }
    void f() {
        System.out.println("Sub");
    }
    public static void main(String args[]) {
        Test t = new Test();
        t.f();
        t.g();
    }
}
```

Questions on Abstract Classes

1. Is it possible to run an abstract class with a main method?
2. Which of the following class definitions define a legal abstract class ? Select all correct answers.
 - a) `class Animal { abstract void growl(); }`
 - b) `abstract Animal { abstract void growl(); }`
 - c) `class abstract Animal { abstract void growl(); }`
 - d) `abstract class Animal { abstract void growl(); }`
 - e) `abstract class Animal { abstract void growl() {System.out.println("growl");} }`
3. Abstract methods cannot be final. True or False?
 1. True
 2. False
4. Abstract methods cannot be static. True or False?
 - a) True
 - b) False
5. Abstract classes cannot have static methods. True or False?
 - a) True
 - b) False
6. Abstract classes cannot have final methods. True or False?
 - a) True
 - b) False
7. A final class cannot have static methods. True or False?
 - a) True
 - b) False
8. A final class cannot have abstract methods. True or False?
 - a) True
 - b) False

Questions on Generics

Question 1:

Generics were introduced in Java version 1.5. Despite this fact, earlier versions still offered the utility classes you learned about (Collection, LinkedList, Comparable, Comparator) but they were non-generic.

1. Briefly describe the differences between the two versions of LinkedList (non-generic and generic).

2. Does one version allow the programmer to do more than the other? If yes, give an example. If no, briefly say why.
3. Informally describe how you could "simulate" a non-generic LinkedList using a generic one.

Question 2:

Suppose you have the following two classes:

```
public class Foo { ... }  
public class Bar extends Foo { ... }
```

1. What type of elements can a LinkedList<Foo> contain?
 - a) Only type Foo
 - b) Only type Bar
 - c) Both type Foo and Bar
 - d) Neither type Foo nor Bar
2. What type of elements can a LinkedList<Bar> contain?
 - a) Only type Foo
 - b) Only type Bar
 - c) Both type Foo and Bar
 - d) Neither type Foo nor Bar
3. What type of elements can a non-generic LinkedList contain?
 - a) Only type Foo
 - b) Only type Bar
 - c) Both type Foo and Bar
 - d) Neither type Foo nor Bar
4. Can you add an element of type Object to LinkedList<Foo> ?