CSc 127B — Introduction to Computer Science II
Fall 2015 (McCann)

Answers to the Practice Questions ahead of Exam #1

The section leaders were kind enough to write some practice questions for you to work as you prepare for the exam; those are in another handout. These are the SLs' suggested answers for the questions.

1. BDE
2. Compile Error
3. Compiles but throws an ArithmeticException
4. Only at Run Time
5. throw
6. catch
7. Throwable
8. If a client can reasonably be expected to recover from an exception, make it a checked exception. If a client cannot do anything to recover from the exception, make it an unchecked exception.
9.

```
public class Circle extends Point
{
        private double radius;

        public Circle(double x, double y, double rad)
        {
                super(x,y);
                radius = rad; //Radii are rad, so there is a deeper truth in our code
        }

        public double getArea()
        {
                return Math.pi * radius * radius;
        }
}
```

10. (a) C
    (b) C
    (c) A
    (d) C
    (e) C
    (f) A
11. Adaptation. Any solution that explains that the functionality of a savings account is closely based upon a BankAccount object would get full credit. This is an example of more than just simple composition, a SavingsAccount object goes further and adapts a BankAccount object.

12. Inheritance and adaptation are occurring here. The extends lets us know that we are inheriting from vehicle and the object truck stored as an instance variable lets us know that some sort of composition is present. When examined further we can see that a moving trucks functionality is based closely off of a truck, thus this composition is adaptation.

13.

```
abstract class Cake {
          int size;
          boolean isDelicious;

          abstract boolean isDelicious();
          abstract Slice takeSlice();
          abstract int slicesLeft();
}
```

14. (a) potential answer: -could talk about Java's single inheritance, and using interfaces to demonstrate multiple inheritance. -could talk about encapsulation, and separating the implementation and the functionality from the user

    (b)

```
public interface Rocket{
    public boolean inFlight = False;
    public static final int NUMBER_OF_WINGS = 4;

    public void liftoff();
    public void countDown();
}
```

    (c)

```
public class SuperSonicRocket implements Rocket{

    public void liftoff(){
        inFlight = True;
    }

    public void countDown(){
        System.out.println(3);
        System.out.println(2);
        System.out.println(1);
    }
}
```

15. Low level binary file I/O is used to write bytes, medium is used to write primitives, and high level is used to write objects. Serializable must be implemented when writing objects. It means that the object can be serialized, It can be represented as a bit string and therefore stored in the file.

16. Text files.

    (a) File arbitraryFileName = new File(practiceTest.txt);

    (b) Yes, you could create a file and write to it using PrintWriter.

    (c) streams of bytes

17. the ObjectType and memory location

18. names, return types, formal parameters

19. formal parameters

20. super

21.

```
Thomas Jefferson and his partner in crime: Aaron Burr. This pairing isn’t a time paradox.
Martin Van Buren and his partner in crime: Aaron Burr. This pairing is a time paradox.
Thomas Jefferson and his partner in crime: Aaron Burr. This pairing isn’t a time paradox.
```

22.

```
       public String toString(){
         String retStr = "";
         for(int i =0; i < ary.length; i++){
           retStr += i + ’\n’
         }
         return retStr;
       }
```