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

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. We'll supply answers in a few days. These questions will be the most valuable if you first study, then sit down and answer the questions in an exam-like setting (quiet location, no computer). If you look at the answers first, you are more likely to say, "Oh, I knew that," even if you really didn't.

1. What will be the output of the program?

   (We didn't cover `finally{}` in class, and it won't be on the exam, but I bet you can guess how it works!)

```
1  public class X
2  {
3      public static void main(String [] args)
4      {
5          try
6          {
7              badMethod();
8              System.out.print("A");
9          }
10         catch (RuntimeException ex)
11         {
12             System.out.print("B");
13         }
14         catch (Exception ex1)
15         {
16             System.out.print("C");
17         }
18         finally
19         {
20             System.out.print("D");
21         }
22         System.out.print("E");
23     }
24     public static void badMethod()
25     {
26         throw new RuntimeException();
27     }
28 }
```

   (a) BD
   (b) BCD
   (c) BDE
   (d) BCDE

2. What happens when we try to compile and execute this program?

```
class Main {
   public static void main(String args[]) {
      try {
         throw 10;
      }
      catch(int e) {
         System.out.println("Got the  Exception " + e);
      }
  }
}
```

   (a) Got the exception 10
   (b) Got the exception 0
   (c) Compile Error

3. What happens when we try to compile and execute this program?

```
class Main {
   public static void main(String args[]) {
      int x = 0;
      int y = 10;
      int z = y/x;
  }
}
```

   (a) Compile Error
   (b) Compiles but throws an ArithmeticException
   (c) Compiles and runs just fine

4. When do exceptions appear?

   (a) Only at Run Time
   (b) Only at Compilation Time
   (c) Can occur at either time
   (d) None of the above

5. Which of these keywords is used to manually throw an exception?

   (a) try
   (b) finally
   (c) throw
   (d) catch

6. Which of these keywords must be used to handle the exception thrown by code inside of a try block?

   (a) try
   (b) finally
   (c) throw
   (d) catch

7. The class Exception extends which Java class?

8. Imagine you are creating a new exception (say, to be added to the API). How can you determine whether your exception should be checked or unchecked?

9. Given below is the code for a Point class:

```
public class Point
{
        private double x;
        private double y;
        public Point(double x, double y)
        {
                this.x = x;
                this.y = y;
        }

        double getArea()
        {
                return 0.0;
        }
}
```

   But, really, Points are boring, theyre nothing but degenerate circles if you really think about it. We in the CS department have no place for such degeneracy, and so we need you to implement an actually useful Circle class. We are also going to be lazy though and utilize all we can from our Point class, all we really need is to have a radius value. Create a new class Circle which inherits from Point, and make sure to override that pesky `getArea()` method with an actual useful `getArea()` method (remember the area of a circle is πr^2).

10. Composition and Adaptation. For each statement below, indicate whether it is a form of composition or of adaptation.

    (a) A bridge object using a bunch of individual wooden plank objects

    (b) A cake object using the flour object

    (c) A red velvet cake object using a cake object

    (d) A bike object using tire objects

    (e) A lamp object using a light bulb object

    (f) A moving truck object using a truck object

11. Let us say we have a BankAccount object. Is a SavingsAccount object simply composed of a BankAccount object or does it adapt a BankAccount object? Explain your reasoning behind picking composition or adaptation to explain the relationship between a BankAccount object and a SavingsAccount.

12. Based on the code below, state what kind(s) of code reuse is or are occurring (inheritance, composition, or adaptation), and why.

```
public class MovingTruck extends Vehicle {

    private Truck truckBase;

    public void car( Truck someTruckBase) {
        truckBase = someTruckBase;
    }

}
```

13. Below you will find the signatures for three different classes of baked goods. Each of the baked goods is beautiful in its own way, but still there is a lot of overlap. So, looking over the following classes I would like you to create an abstract superclass Cake, from which these all inherit. Cakes should contain the signature for as many methods as there is overlap, but no further. Remember, the point of abstract classes is to capture as much commonality between subclasses as possible, without too strictly defining the functioning of the lower classes.

```
1  public class ChocolateCake
2  {
3             private int size;
4             private boolean isTasty;
5             private Enum typeOfChocolate;
6
7             public ChocolateCake(int size, boolean isDelicious, Enum typeOfChocolate);
8             public boolean isDelicious();
9             public Enum whatChocolate();
10            public Slice takeSlice();
11            public int slicesLeft();
12 }
13
14 public class CarrotCake
15 {
16            private int size;
17            private boolean isTasty;
18            private boolean hasRaisins;
19            private boolean hasNuts;
20
21            public CarrotCake(int size, boolean isDelicious,
22                              boolean hasRaisins, boolean hasNuts);
23            public boolean isDelicious();
24            public boolean hasRaisins();
25            public boolean allergenAlert();
26            public Slice takeSlice();
27            public int slicesLeft();
28 }
29
30 public class Muffin
31 {
32            private int size = 1;
33            private boolean isTasty;
34            private boolean hasNuts;
35
36            public Muffin(boolean isDelicious, boolean hasNuts);
37            public boolean isDelicious();
38            public boolean allergenAlert();
39            public Slice takeSlice();
40            public int slicesLeft();
41 }
```

So remember, we're trying to find the maximum overlap of methods to specify in our abstract class to capture similar behavior. Don't worry about writing the method code itself. Just write the abstract class and the headers we need for this.

14. Interfaces.

    (a) Describe the benefit of using interfaces.

    (b) What is wrong with the interface given below? Rewrite the interface to correct the issues.

```
1  public interface Rocket{
2
3      public boolean inFlight = False;
4      public static final int NUMBER_OF_WINGS = 4;
5
6      public void liftoff(){
7          inFlight = True;
8      }
9
10     public void countDown(){
11         System.out.println(3);
12         System.out.println(2);
13         System.out.println(1);
14     }
15
16 }
```

    (c) Create a class called SuperSonicRocket that implements the corrected interface from part(b).

15. Describe the difference between low, medium, and high Binary File levels. What interface must be implemented when writing objects to Binary Files? Why?

16. Text files.

    (a) In order for a Java program to learn the size of a file named `practiceTest.txt`, what is the first line of code it would need to execute?

    (b) Could I type up a term paper for my English Literature course using only a Java program? If no, why not? If yes, how?

    (c) The readers and writers we have been using in class to work with files rely on the fact that Java sees files as _____ of _____ .

17. Oftentimes when we write our own classes, the toString() needs to be overridden to return any useful information. If it is not, and toString() is called on an instantiated Object what will Java return?

    (a) the index of the Object

    (b) it will throw an exception

    (c) the ObjectType and memory location

    (d) the argument of toString()

18. To write a method to override another method they must have the same three components. What are they?

19. Overloading occurs when the method has 2 of 3 properties of an overridden method. Which one changes?

20. You can "override the override" to a method by using what in front of the method call?

    (a) this

    (b) Object

    (c) next

    (d) super

21. Look at the class below. What will the output of its main method be?

```
1  public class President
2  {
3      private String name;
4      private String vP;
5      private boolean timeParadox;
6
7      public President(String name, String vP)
8      {
9          this.name = name;
10         this.vP = vP;
11         timeParadox = false;
12     }
13
14     public void vPSteal(President pArg)
15     {
16         vP = pArg.vP;
17         timeParadox = true;
18     }
19
20     public String toString()
21     {
22         String temp;
23         if (timeParadox)
24             temp = "is";
25         else
26             temp = "isn't";
27         return name + " and his partner in crime: " + vP + ". This pairing "
28                 + temp + " a time paradox.";
29     }
30
31     public static void main(String[] args)
32     {
33         President[] p = new President[3];
34         p[0] = new President("Thomas Jefferson", "Aaron Burr");
35         p[1] = new President("Martin Van Buren", "Richard Mentor Johnson");
36         p[2] = new President("William Howard Taft", "James S. Sherman");
37
38         p[2].vPSteal(p[0]);
39         p[1].vPSteal(p[2]);
40         p[2] = p[0];
41
42         for (int i = 0; i < p.length; i++)
43         {
44             System.out.println(p[i]);
45         }
46     }
47 }
```

22. To the class below, add a toString() method that overrides Object's and returns a String composed of each element of the array such that, when the string is printed, each value of the array will appear on a new line.

```
1   class NumAry{
2       int[] ary = {1,1,2,3,5,8,13};
3
4
5
6   }
```