Comp 170 Final Exam Overview.

Exam Ground Rules
The exam will be closed book, no calculators. You may bring notes on two sides of 8.5x11 inch paper (either both sides of one sheet, or two sheets written on single sides AND the 4-page javaNotes doc.

I have the review set up as all with Java syntax.

Main topics:

1. Concepts of loops, decisions, variable updates, function parameters and return values, constructors, instances, methods, largely common with Python,
2. Java variable declarations, necessity of classes
3. Difference of Java / from Python
4. Be able to at least look up Java string manipulation syntax, and then follow it
5. Arrays: declaring, creating, initializing, length
6. Method name overloading
7. Interface: syntax for writing and using
8. Java syntax for instance variables, constructors, methods, toString, static functions and variables
9. Some HashMap and ArrayList stuff, but ideas common to Python, with translation syntax from the javaNotes
10. Java String format with fieldwidth and double variable precision; printing, like
    `System.out.format("Blah Blah %5s %7s %7.2f%n", "Hi", 123, 6.2345);`
11. Scanner from System.in, and reading from a file, though I will not ask you to initialize a Scanner from a file with all the file initialization verbiage needed in Java!
12. Follow code to play computer, including arbitrary use of arrays and nested loops where you do not already know what is supposed to happen!
13. Read and write object code, including methods where objects of the receiver's type are parameters.

The newest, most different stuff is Java variable declaration, array syntax, class instance syntax. Expect those to be emphasized.

Same basic instructions for studying before doing review problems as before, but now I am letting you look up a lot of Java syntax.

Review problems start on the next page, then followed by my solutions.

Sample Review Problems for Exam 3

1. What is printed by this code fragment?

```
double a = (7/2)*4.0;
double b = (8.0/4)*10;
System.out.format("%.1f %.1f", a, b);
```

2. What is printed by this program fragment? Show blanks as boxes.

```
System.out.println("0123456789");
int x = 1;
for (int n = 0; n < 4; n++) {
  System.out.format(":%5.1f%n", x/3.0);
  x *= 20;
}
```

3. What is printed by this program fragment? Show blanks as boxes.

```
System.out.println("123456");
for (int n = 1; n < 4; n++) {
  String fString = ":%" + n + "s%n";
  System.out.format(fString, 2*n);
}
```

4. Consider the following code fragment:

```
HashMap<String, String> m =
          new HashMap<String, String>();
m.put("a", "stuff");
m.put("b", "junk");
m.put("c", "garbage");
System.out.println(m.get("b"));
Set k = m.keySet();
```

a. What is printed? b. What are the elements of the Set `k`?

5. What does the Test program display? (**See Foo -->**)

```
class Test
{
   public static void main(String[] arg) {
      Foo f1 = new Foo(3, 5);                          //1
      Foo f2 = new Foo(2, 10);                         //2
      System.out.println("f1:" + f1 + " f2: " + f2);    //3
      f2.pah(f1);                                      //4
      System.out.println("f1:" + f1 + " f2: " + f2);    //5
   }
}
```

```
class Foo
{
   private int x, y;

   public Foo(int a, int b) {     //1
      x = a;                      //2
      y = b;                      //3
   }

   public String toString() {
      return  "x:"+x+ " y:" +y;   }

   public void pah(Foo f) {       //1
      x = x + f.y;                //2
      y = x*f.x;                  //3
   }
}
```

6. Write the Java code for an interface Bar which includes just one method, with signature:
public void foo`(int x)`

7. Remember the Fraction class with int instance variables num and den. Complete the mult method for the class Fraction.

```
/**
 * Multiply this Fraction by another.
 * @param  f what this Fraction is multiplied by.
 * @return the product of this Fraction and f.
 */
public Fraction mult(Fraction f)  // like *
```

8. List is an interface satisfied by ArrayList. Which of the following two declarations and initializations make sense, individually? Explain.

a. `List<String> list = new ArrayList<String>;`

b. `ArrayList<String> list = new List<String>;`

9. Complete these methods.

```
public static void aSqr(int x[])
// squares the elements of x:
// For example if x initially contains 2, 3, 5, 7,
//         then at the end x contains  4, 9, 25, 49.

public static int aProd(int x[])
// returns the product of the elements of x (or 1 if there are no elements):
// For example if x initially contains 2, 3, 5,
//    then aProd returns 30.

public static int nMatch(int x[], int y[])
// returns the number of matches of elements of x and y
//    with the same subscript:
// For example if  x contains 2, 3, 5, 7, 9
//                 y contains 2, 7, 5, 8, 9, 6, 3
//    then nMatch returns 3  (2's, 5's, 9's, not 7's or 3's)
```

10. Consider the class P10 with an instance variable x which is an int array. First write a constructor that takes an array parameter and makes x be a *copy* of the parameter. Then modify the static methods of problem 9 to be instance methods with the headings below.

```
class P10
{
      private int[] x;

      public P10(int[] a) // creates x and copies the elements of a into it
      public void aSqr() // replace elements of x by their squares
      public int aProd() // return product of elements of x
      public int nMatch(P10 y)//count matches between this.x and y.x
      ...
}
```

11. Complete the code for the method readFractions, below. Suppose the Scanner parameter is connected to a text file containing pairs of integers, which are intended as numerators and denominators of fractions. For instance, if the Scanner is connected to a file containing

```
2 5
-2 11
8 3
```

then the method should return a new ArrayList of Fractions containing 2/5, -2/11, and 8/3.

```
public static ArrayList<Fraction> readFractions(Scanner in)
```

12. Complete the method shifted, which takes an array v as parameter and returns a new array of the same size, where the last element of v becomes the first element of the returned array, and all the other elements of v are shifted one index in the returned array. Examples showing the elements of v and the elements of the returned array:

v: 1, 2, 3, 4, 5	v: 2, 4, 6, 8	v: 2, 5	v: 7
returned: 5, 1, 2, 3, 4	returned: 8, 2, 4, 6	returned: 5, 2	returned: 7

```
public static int[] shifted(int[] v)
```

13. Write a function to print a triangle of asterisks ('*'), starting with one asterisk, building up to n asterisks. Examples: for n=4: * n=2: * n= 1: *

```
**
***
****
```

```
public static void printTriangle(int n)
```

14. The state of an object of the Java class FruitTree is specified by a single integer, its number of branches. A new FruitTree has exactly one branch. A FruitTree can grow, meaning it adds exactly one branch. The grow method returns nothing. A FruitTree has a method produce, which returns the number of fruit produced by the current number of branches. If the tree has n branches, the number of fruit is n(n-1)/2.
A FruitTree object has a toString method that displays the number of branches, for instance "Branches: 3".

The following sequence should make sense. Write a class definition that is consistent:

```
FruitTree ft = new FruitTree();
System.out.println(ft);            // Branches: 1
ft.grow();                         // (now two branches)
System.out.println(ft.produce()); // 1  2(1)/2 = 1
ft.grow();
ft.grow();
System.out.println(ft);            // Branches: 4
System.out.println(ft.produce()); // 6  4(3)/2 = 6
```

15. What does this program print?

```
public class Prob15 {

    public static void main(String[] args)
    {
        print(3, "now");
        print("now", 5);
    }

    public static void print(String a, int b)
    {
        System.out.println(a + " and " + b);
    }

    public static void print(int a, String b)
    {
        System.out.println(a + "; " + b);
    }
}
```

// SOLUTIONS ON NEXT PAGE

Final Exam Review problem solutions

1. 12.0 20.0
a = 3*4.0 = 4.0 (initial *integer* division)
b = 2.0*10 = 20.0 (display with 1 decimal place)

2.
```
0123456789
:  0.3
:  6.7
:133.3
:2666.7   // squeezes beyond format field size
```

3.
```
123456
:2          format string ":%1s%n" prints value of 2*n: 2 in 1 column
: 4         format string ":%2s%n" prints value of 2*n: 4 in 2 columns
:  6        format string ":%3s%n" prints value of 2*n: 6 in 3 columns
```

4. a. junk b. "a", "b", "c" (order *not* specified)

5.
```
f1: x:3 y:5 f2: x:2 y:10
f1: x:3 y:5 f2: x:7 y:21
```

line by line: receiver (this) in boldface

method	Test:main----				Foo:Foo		Foo:bar		
object	f1		f2				f		
variable	x	y	x	y	a	b	x	y	
line									comments
Test 1									invoke Foo
Foo 1					3	5			
Foo 2	**3**				3	5			new Foo will be called f1, so I write to that object, assigning to x
Foo 3	**3**	**5**			3	5			similarly, for y
Test 2									invoke Foo
Foo 1	3	5			2	10			
Foo 2	3	5	**2**		2	10			new Foo will be called f2, so I write to that object, assigning to x
Foo 3	3	5	**2**	**10**					similarly, for y
Test 3.	3	5	2	10					calls toString for f1, f2 returning "x:3 y:5" and "x:2 y:10";adds f1:, f2:
Test 4.	3	5	2	10					to method pah with receiver f2
pah 1	3	5	2	10			3	5	f is an alias for f1. Since the receiver is f2, x and y refer to f2.x and f2.y
pah 2	3	5	**7**	10			3	5	2 + 5 = 7 to x
pah 3	3	5	7	**21**			3	5	7*3 = 21 to y (using new x!)
Test 5.	3	5	7	21					calls toString for f1, f2 returning "x:3 y:5" and "x:7 y:21"; adds f1:, f2:

6.
```
interface Bar
{
  public void foo(int x);
} // note the semicolon above!
```

7.
```
{
  return new Fraction(num*f.num,
                      den*f.den);
}
```

8a. Wrong: List is an interface. Only real classes have constructors. Also, FYI, all Lists are not ArrayLists.

8b. Legal: An ArrayList is a kind of List.

9.

```
public static void aSqr (int[] x)
{
   for (int i = 0; i < x.length; i++)
      x[i] *= x[i];
}
```

```
public static int prod(int[] x)
{
  int p = 1;
  for (int i = 0; i < x.length; i++)
      p *= x[i];
  return p;
}
```

```
static int nMatch(int[] x, int[] y)
{
  int matches = 0;
  int n = x.length;
  if (n > y.length)  n = y.length;
  for (int i = 0; i < n; i++)
    if (x[i] == y[i])
      matches++;
  return matches;
}
```

10.

```
public P10(int[] a)
{ // creates x and copies the elements of a into it
  x = new int[a.length];
  for (int i = 0; i < x.length; i++)
    x[i] = a[i];
}
```

```
public void aSqr()  //   no changes in body from #9
public int aProd()   //no body change from #9
```

public int nMatch(P10 y)
just replace y by y.x everywhere in problem 9, so
y[i] is replaced by y.x[i];
y.length is replaced by y.x.length

11.

```
{
  ArrayList<Fraction> list = new ArrayList<Fraction>();
  while (in.hasNextInt())
    list.add(new Fraction(in.nextInt(), in.nextInt()));
  return list;
}
```

12.

```
{
  int[] a = new int[v.length]
  for (int i = 0; i < v.length-1; i++)
      a[i+1] = v[i];
  a[0] = v[v.length - 1]
  return a;
}
```

13.

```
{
  for (int i = 1; i <= n; i++) {
    for (int j = 0; j < i; j++)
      System.out.print('*');
    System.out.println();
  }
}
```

14.

```
public class FruitTree
{
   private int br;  // branches

   public FruitTree()
   {
      br = 1;
   }

   public void grow()
   {
      br++;
   }

   public int produce()
   {
       return br*(br-1)/2;
   }

   public String toString()
   {
      return "Branches: " + br;
   }
}
```

15. Note overloading of the name `print`!

```
3; now
now and 5
```