Joanna Klukowska
joannakl@cs.nyu.edu

Practice Questions for Final Exam: Advanced Java Concepts + Additional Questions from Earlier Parts of the Course

1. Given the following hierarchy:

```
class Alpha { ... }
class Beta extends Alpha { ... }
class Gamma extends Beta { ... }
```

   In what order are the constructors for these classes called when a `Gamma` object is instantiated?

2. What class is a superclass of every other class? What does one need to do to use methods inherited from that class.

3. A superclass reference can refer to a subclass object. True or False?

   A subclass reference can refer to a superclass object. True or False?

   Show an example of the true scenario? (Do not define classes, just state which class inherits from which and then write a statement that illustrates one or both of the cases above.)

4. Explain what `extends` keyword is used for.

5. Show how to open a file for reading characters.

6. How many classes can extend a superclass? How many superclasses can a class extend?

7. Show the output of following program:

```
public class Test {
  public static void main(String[] args) {
    A a = new A(3);
  }
}
class A extends B {
  public A(int t) {
    super();
    System.out.println("A's constructor is invoked");
  }
}
class B {
  public B() {
    System.out.println("B's constructor is invoked");
  }
}
```

8. Answer the following multiple choice questions. There might be more than one correct answer - mark all that apply.

   (a) A subclass inherits ____________ from its superclass.

      i. private methods

Joanna Klukowska
joannakl@cs.nyu.edu

ii. protected methods
iii. public methods
iv. static methods

(b) When you implement a method that is defined in a superclass, you ________ the original method.

i. overload
ii. override
iii. copy
iv. call

(c) What is the output of running the class C.

```
public class C {
  public static void main(String[] args) {
    Object o1 = new A();
    Object o2 = new B();
    System.out.print(o1);
    System.out.print(o2);
  }
}
class A extends B {
  public String toString() {
    return "A";
  }
}
class B {
  public String toString() {
    return "B";
  }
}
```

i. AB
ii. BA
iii. BAB
iv. ABA
v. None of above

(d) If a class named `Student` has a constructor `Student(String name)` defined explicitly, the following constructor is implicitly provided.

i. `public Student()`
ii. `protected Student()`
iii. `private Student()`
iv. `Student()`
v. `None`

9. Design a class named `Person` and its two subclasses named `Student` and `Employee`. A person has a name and email address. A student has a GPA. An employee has an office number and a salary. Override the `toString` method in each class to display the class name and the person's name.

10. Consider the following code.

```
public class Division {

   public static void main(String[] args) {
       Scanner in = new Scanner (System.in);
```

```java

        int num1 = 0, num2 = 0;

        try{
            System.out.println("Please enter two integers to divide:");
            System.out.print("numerator: ");
            num1 = in.nextInt();
            System.out.print("\ndenominator: ");
            num2 = in.nextInt();

            System.out.printf ("The quotient of %d and %d is %d\n",
                                num1, num2, num1/num2);

        }
        catch (ArithmeticException e) {
            System.out.println("Sorry, cannot divide by zero!\n");
        }
        catch (InputMismatchException e ) {
            System.out.println("This is not an integer!\n");
        }
        catch( Exception e ){
            System.out.println("You did something wrong!\n");
        }

        System.out.println("Thank you for trying Division. ");
        in.close();
    }

}
```

What is the output when user enters

(a) 10 and 5
(b) 12 and 5
(c) 12 and 0
(d) 11 and 5.5

11. Write a `CalendarDate` class that has two private data fields `day`, and `month`. Your class should have
    - a default constructor that sets the date to a random day in May (May has 31 days)
    - compareTo method (remember that it returns 0 when two CalendarDate objects are equal, a number < 0 when this object is smaller than the parameter object, and a number > 0 when this object is greater than the parameter object)

12. Write a `checkDate()` method that given an array of `CalendarDate` objects (as defined in question 11) and another `CalendarDate` object returns `true` if the object is in the array, and `false` otherwise.

13. Write a program that opens the file European_capitals.txt whose content is reprinted below:

```
Paris 2243833
Amsterdam 1108297
Warsaw 1715517
Rome 2645907
Prague 1262106
London 8308369
Berlin 3415091
```

The file contains names of several capitals and their population size. You can assume that there is a single space between the name of the city and the number. You can assume that each city is listed on a new line. Your program should read in the data from this file into two arrays (one of type String, the other of type int).

You program should display the following information:

- name of the city with smallest population
- name of the city with largest population
- total population size for all the cities.

Make sure that the two arrays remain in sync as you are doing this.

14. For the purpose of this question, assume that you are developing static methods for your own statistics library `MyStats` (it is similar to the Java's `Math` library that you have been using throughout the semester).

    (a) Write a public static method `threeEqual()` for `MyStats` that takes three int values as arguments and returns `true` if all three numbers are equal, `false` otherwise.

    (b) Write a public static method `median()` for `MyStats` that takes as an argument an array of sorted integer values and returns the middle value. Your method should first verify if the array is sorted, and if it is not it should throw
    an `IllegalArgumentException`.

    (c) Give a single Java expression that a client of `MyStats` could use to test whether the medians of three arrays of integer values `int[] a`, `int[] b`, and `int[] c` are all equal.

15. Consider the following program.

```java
public class Mystery {
    public static void main(String[] args) {

        int N = args.length;
        String[] a = new String[N * 2];
        for (int i = 0; i < N; i++) {
            a[i] = args[i];
            a[i + N] = args[N - i - 1];
        }
        for (int i = 0; i < a.length; i++)
            System.out.println(a[i] + " ");
        System.out.println();
    }
}
```

    (a) What does this program print out when the following command is executed (from the command line)?

    `java Mystery aaa bbb ccc`

    (b) What does this program print out when the following command is executed (from the command line)?

    `java Mystery xxxx yyyy`

16. Consider the following program, which is supposed to read in integer N from standard input, read N strings from standard input, and print them to standard output in reverse order.

```java
public class ReverseInputBuggy
{
    public static void main(String[] args)
```

Joanna Klukowska
joannakl@cs.nyu.edu

```
 4    {
 5        java.util.Scanner in = new java.util.Scanner(Std.in);
 6        int N = in.nextInt();
 7        String s;
 8        for (int i = 1; i < N; i++)
 9            s[i] = in.next();
10        for (int i = N; i >= 0; i--)
11            System.out.println(s[i]);
12    }
13 }
```

This program has three bugs.

(a) Which bug prevents the program from compiling successfully? Identify the line number where the bug appears and give a correct version of this line of code.
Line number ______
Correct version:

(b) After fixing the first bug, which bug causes the program to crash? Identify the line number where the bug appears and give a correct version of this line of code.
Line number ______
Correct version:

(c) After fixing the first two bugs, which bug causes the program to produce incorrect output? Identify the line number where the bug appears and give a correct version of this line of code.
Line number ______
Correct version:

17. Implement the Java class `Digits`, which allows you to access a number by its individual digits. Here is the complete interface:

```
public class Digits {

    // create Digits version of num
    // assume num >= 0
    public Digits(int num) { . . . }

    // gets digit i of the number,
    // such that digit 1 is the left-most digit,
    // and digit numDigits() is the right-most  digit
    // assue 1 <= i <= numDigits()
    public int getDigit(int i) { . . . }

    // returns the number of digits in the number
    public int numDigits() { . . . }

    // returns the integer itself
    public int getInt() { . . . }

}
```

Here's an example of using the class:

```
Digits zero(0);
System.out.println(zero.numDigits() + ‘‘ ‘‘ + zero.getDigit(1));
// prints:  1 0
```

```
Digits digits(5207);
System.out.println(digits.numDigits()); // prints:  4
System.out.println(digits.getDigit(1) + ‘‘ ‘‘ + digits.getDigit(3));
// prints:  5 0
System.out.println(digits.getInt()); // prints:  5207
```

Hint: the modulus (%) and integer division operators will be useful in your implementation.

18. Write the Java function `lengthOfSortedSequence` which returns the length of the longest sorted sequence in an array. For this problem a sorted sequence is a sequence of non-decreasing values. Here are some examples:

`array`	return value from `lengthOfSortedSequence( array )`
[-7 3 99 -10 0 0 43 10 20 30 5]	4
[-1, 2, 3, 3]	4
[5, 2, 1]	1
[1]	1
[]	0

19. Write a Java class `Point` that represents (x,y) point in a plain. The class should implement `Comparable` interface. The points should be compared based on their distance from the origin (point (0,0)). The distance from the origin can be computed using

$$distance = \sqrt{x^2 + y^2}.$$

Your class should implement all methods needed for the following code to compile and run successfully:

```
Random r = new Random;
Point [] myPoints = new Point[10];
for (int i = 0; i < myPoints.length; i++)
   myPoints[i] = new Point(r.nextDouble(), r.nextDouble() );
Arrays.sort(myPoints);
```

You do not need to provide any additional methods.

20. Implement the static method `mode`, which returns the value that occurs most often in an array of integers. You may assume the array has at least one element, and all of the values in the array are in the range `0 - 100`. If there is more than one value that occurs most often, return the smallest such value (See Example 2 below). Here are some examples:

Example 1: array contains 99 86 99 99 95 86, `mode(array)` returns 99
Example 2: array contains 23 15 15 74 23, `mode(array)` returns 15
Example 3: array contains 76, `mode(array)` returns 76

For full credit, your answer must only traverse the data in array once**.** Hint: you are allowed to use additional memory.

21. For each of the following Java definitions, fill in the question marks (???) such that the given main class always prints the number 42. If it's not possible to do this, then explain why. Keep your answers as simple as possible.

(a)
```
class C {
    public int foo() {
        return 10;
    }
}
```

Joanna Klukowska
joannakl@cs.nyu.edu

```
class S extends C {
        ???
    }
class Main {
    public static void main(String[] args) {
        S x = new S(42);
        System.out.println(x.foo());
    }
}
```

(b)
```
class C {
    private int x;
    public C(int x) {
        this.x = x;
    }
    public int foo() {
        return x;
    }
}
class S extends C {
    public S() {
        ???
    }
}
class Main {
    public static void main(String[] args) {
        S x = new S();
        System.out.println(x.foo());
    }
}
```

(c)
```
class C {
    ???
}
class Main {
    public static void main(String args[]) {
        C x = new C();
        C y = new C();
        System.out.println(x.get() + y.get() + 1);
    }
}
```

22. Write the method `sumSeries()` that given an integer argument `n` will return as a double the following sum:

$$\frac{1}{n} + \frac{2}{n-1} + \frac{3}{n-2} + \ldots + \frac{n-1}{2} + \frac{n}{1}.$$

Do not write a full program with input and output, just the method.

Hints/Notes:

- You need only a single loop.
- The return value is a `double`, make sure that as the sum is computed using double division and not integer division.

- You don't need to check for valid values of n, assume it's an integer > 0.

23. Define a class `PongBall` that is used for a `Pong` game (just like the one you implemented for one of the assignments). The class should have data fields representing its x and y coordinates of the center, and two data fields that indicate increments in both x and y directions when the ball moves. It should have a constructor that initializes the balls position to a random location within a window that is 300x300 pixels - you need to decide if this constructor requires any parameters or not. The increments in both x and y directions should be set to 2. Provide a move() method that adjusts the balls position (you do not need to worry about the boundary conditions) and draws the ball in its new position in the window.

24. What would be printed by the following programs?

(a)
```
public class CatsAndDogs {

    public static void main(String[] args) {
        foo("Cats and Dogs", 4);
    }

    public static void foo ( String s, int n ) {
        if (n <= 1)
            System.out.println("Cats");
        else {
            System.out.println( s ) ;
            foo ( s, n-1 );
        }
    }
}
```

(b)
```
public class Numbers {

    public static void main(String[] args) {
        int [] list = {1, 2, 3, 4, 5};

        System.out.println( foo (list , 0, list.length-1) );
    }
    public static int foo ( int [] nums, int begin, int end ) {
        if ( begin == end )
            return nums[begin];
        else
            return nums[begin] + foo(nums, begin+1, end);
    }
}
```

25. Write an iterative and recursive implementation of a method that given two values: a real number x and an integer y, computes the value of x^y.

26. Given the following recursive method, answer the short questions below:

```
int puzzle (int base, int limit )
{
  if ( base > limit )
    return -1;
  else if (base == limit)
    return 1;
  else
```

Joanna Klukowska
joannakl@cs.nyu.edu

```
8      return base * puzzle( base + 1, limit) ;
9 }
```

What would be printed by the following calls to the recursive method(write your answer next to each statement):

```
System.out.println( puzzle (5,2) );
System.out.println( puzzle (2,5) );
System.out.println( puzzle (0,0) );
```

Which lines of code contain the base case(s)?

Which lines of code contain the general/recursive case(s)?

What is the total number of calls made to `puzzle` method when calculating `puzzle (5,2)` (include the initial call)?

What is the total number of calls made to `puzzle` method when calculating `puzzle (2,5)` (include the initial call)?

27. What is the problem with the following recursive method to compute a^n , for positive integer n?

```
public static double exponent(double a, int n) {
    return a*exponent(a,n-1);
}
```

Describe briefly how this should be fixed. Your solution should still use recursion.