Name:_____

CSCI-UA.0101-004 – Midterm Exam #1

March 3rd, 2015

Instructor: Joseph Versoza

Ask the person to your left for their first name (leave blank if next to empty seat or wall):

_____

Ask the person to your right for their first name (leave blank if next to empty seat or wall):

_____

Keep this test booklet closed until the class is prompted to begin the exam

- Computers, calculators, phones, textbooks or notebooks are **not allowed** during the exam
- Please turn off your phone to avoid disrupting others during the exam

Instructions

- The code samples use an abbreviation, `syso`, for `System.out.println`
- You can use the same abbreviation when writing your own code.
- Depending on the question, you may or may not need to define a class and / or a main method. Read the instructions carefully.
- Character codes can be found on the front of the last page.
- The last page can be used as scratch paper.

1. Name 4 methods that you can call on a String object. (2 points)

   equals, charAt, substring, concat, indexOf, toUpperCase, etc.

2. What is the result of the following boolean expressions? (2 points)

   (a) **false** `(5 > 2 ^ 2 < 5)` (b) **true** `(! Character.isUpperCase('a'))`

   (c) **true** `(5 > 2 || 2 < 5)` (d) **true** `(false && true || 5 != 2)`

3. Answer the **questions about** the **code on the left** in the columns on the right. (6 points)

   (a) Assume that all of the code is within the **`main`** method of a Java program
   (b) Additionally, assume that a **`Scanner`** object called **`input`** was declared and created earlier in the program
   (c) **Show your work** (where possible) **for partial credit** (for example, write the values of variables, add comments in the code inline, etc.)

Code	Question #1	Question #2

`int x = 1, y = 0;`
 	
`while(x > y && x >= 0 && y >= 0)`
`do {`
 	
`  syso("Please enter two numbers");`
 	
`  x = input.nextInt();`
 	
`  y = input.nextInt();`
 	
`} while(x > y && x >= 0 && y >= 0);`
 	
`// or do if and break`
`syso("Done!");`
	Give two sets of **user input** (that is two sets of two numbers) that will **cause this `do-while` loop to stop**.
	
	5 7
	10 -1
		Convert the `do while` loop into a ***regular*** **`while`** loop. You can mark-up the code on the left (cross out, add lines, draw arrows) to do this.

`int total = 0;`
`for(int i = 40; i > 10; i -= 10) {`
`  syso(i);`
`  total += i;`
`}`
`syso(total);`
	When would you use a `for` loop over a `while` loop?
	When you know how many iterations there are, or if you're iterating over an Array (or some other iterable)
		What is the **output** of this program? **Show** your **work** for **partial credit**.
		
		40
		30
		20
		90

`int num = input.nextInt();`
`String weekday = "";`
`switch(num) {`
`  case 1:`
`  weekday = "Monday";`
`  case 2:`
`  weekday = "Tuesday";`
`  case 3:`
`  weekday = "Wednesday";`
`  case 4:`
`  weekday = "Thursday";`
`  case 5:`
`  weekday = "Friday";`
`}`
`syso(weekday);`
	What is the **output** of the code if the user enters the number, **3**?
	
	Friday (does not print everything out!)
		The program is meant to display the weekday corresponding the number, num... for example, `1` should result in `Monday`. **There is a logical error** in the code. **Fix the error** by marking-up the code on the left (cross out, add lines, draw arrows).
		
		Place break at end of each case

4. Answer the questions about running Java programs below. (3 points)

   (a) In Java, we usually specify a `main` method in our class. What is the `main` method? Does it **return** a **value**? What **argument**(s) does it take?

   The main method is the method that the JVM invokes when a java program is run. It does not return a value, and it takes an Array of Strings as an argument.

   (b) What is the difference between a `.class` file and a `.java` file? Describe what is contained in each.

   The .java file is the source file – it contains your actual code. The .class file is the compiled file. It contains bytecode.

   (c) What is the **JVM** (what does it stand for, what is it used for, etc.)? What is **bytecode**?

   The JVM is the Java Virtual Machine. It's an interpreter – it runs Java bytecode. Java bytecode is *machine code* for the Java Virtual Machine. Java source files are compiled to bytecode.

5. Write a **method** called **`lastIndexOf`**. It will give back the index of the last occurrence of a `String` in an `Array` of `Strings`. The header should match the specification below. (7 points)

```
public static int lastIndexOf(String[] haystack, String needle)
```

   (a) `haystack` is the `String Array` to search, `needle` is the `String` to search for
   (b) give back the index of the last occurrence of `needle` in `haystack` (`String` in `Array`)
   (c) give back `-1` if needle is not in haystack (`String` is not found in `String Array`)
   (d) example usage below:

```
String[] words = {"hello", "hi", "what's up?", "hi", "yo"};
lastIndexOf(words, "hi"); // --> 3
lastIndexOf(words, "hello"); // --> 0
lastIndexOf(words, "bye"); // --> -1
```

```
public static int lastIndexOf(String[] haystack, String needle) {
      int index = -1;
      for(int i = haystack.length - 1; i >= 0; i--) {
            if(haystack[i].equals(needle)) {
                  index = i;
                  break;
            }

      // no need for else continue for no-op
      // if you use else to set to -1... watch out!


      }
      return index;
}
```

6. Given the following `Array`, `int[] numbers = {1, 1, 2, 3, 5, 8}`, print out every element in the Array using a for loop and a for each loop (do not use the Arrays.toString method). (2 points)

for loop	**for each loop**

`for(int i = 0; I < numbers.length; i++) {`
`  syso(numbers[i]);`
`}`
	`for(int i: numbers) {`
	`  syso(i);`
	`}`

7. Write a **complete program**, **including** the **class definition**,the **`main`** method definition, and any necessary **imports**. The program should output the **numeric sum of all of the digits in a `String`** provided by the user. (9 points)

   (a) ask the user for a `String`
   (b) add all of the numbers that are contained in the String; ignore everything else
   (c) output the total
   (d) **hint**: you can do this with or without unicode code points
   (e) **hint**: similar to the `BinHexDec.java` homework, you can use `Character.getNumericValue` to get the int value that a single character represents – example usage of the method is as follows: `Character.getNumericValue('5') // returns 5`
   (f) **hint**: alternatively, you can use Integer.parseInt, but you'll have to figure out how to extract a single string from the user input
   (g) sample user interaction below:

```
Please enter a string:
> 12hello3
The total of the digits in the string is 6
```

```
import java.util.Scanner;


public class SumDigits {
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.println("Please enter a string:");
		String s = input.next();
		char ch;
		int total = 0;
		for(int i = 0; i < s.length(); i++) {
			ch = s.charAt(i); // extract character
			if(Character.isDigit(ch)) { // check if digit or use code points
				total += Character.getNumericValue(ch);
			}
		}
		System.out.printf("
				The total of the digits in the string is %s", total);
	}
}
```

8. Write the output of the following code in the space provided. (3 points)

`// 1 point`
`char ch = 'a';`
`syso(ch++);`
`syso(ch);`
 	
`a`
`b`
	`// 1 point`
	`int x = 10;`
	`syso(--x);`
	`syso(x);`
	
	`9`
	`9`

`// ½ point`
`int y = 20.0;`
`syso(m / 10);`
 	
`error`
	`// ½ point`
	`double z = Math.PI;`
	`System.out.printf("%.1f", z);`
	
	`3.1`

9. Write a program that **continually generates a random** number from **1 through 13** until the sum of all numbers is greater than 21. Assume that a `main` method already exists and that your code is within the `main` method. (7 points)

   (a) It will continue to do this until the total of all numbers generated is over 21.
   (b) The numbers **11 through 13** will count as **adding 10** to the **total**.
   (c) Print out each number generated; if the number is **11 through 13**, print out 10 in parentheses after the number.
   (d) Print out the total at the end... example output below:

6	6	12 (10)
7	2	3
11 (10)	7	2
Total: 23	10	11 (10)
	Total: 25	Total: 25

```
int total = 0;
int num;
while (total <= 21) {
      num = (int) (Math.random() * 13) + 1;
      if(num >= 11) {
            System.out.printf("%s (%s)\n", num, 10);
            total += 10;
      } else {
            System.out.println(num);
            total += num;
      }
}
System.out.println("Total: " + total);
```

10. Write a **method** called **`shift`**, which will give back a new `Array` by **shifting** all of the **elements** in the original Array **to the left or to the right**. You do not have to specify the class that this methods is in, and you do not have to write a a `main` method (assume both already exist). (10 points)

(a) it should have **two parameters**, an **`Array of ints`**, and an **`int`** representing an offset
(b) it should **return** a new **`Array`**
(c) assume that the `Array` passed in is not empty
(d) assume that the absolute value of the offset will be less than the `Array`'s length and that the Array will have at least 2 elements
(e) if the offset is positive, shift every element to the right for that many positions
(f) if the offset is negative, shift every element to the left for that many positions
(g) if an element is shifted *out* of the `Array`, move it to the beginning or the end of the `Array`
(h) example output below – `Array` and offset are on the left, `Array` returned is on the right after `-->`

```
[1, 2, 3, 4, 5], 1  --> [5, 1, 2, 3, 4] // shift all elements 1 to the right
[1, 2, 3, 4, 5], 2  --> [4, 5, 1, 2, 3] // shift all elements 2 to the right
[1, 2, 3, 4, 5], -2 --> [3, 4, 5, 1, 2] // shift all elements 2 to the left
```

```
// v1

public static int[] shift(int[] arr, int offset) {
      int[] shifted = new int[arr.length];
      for(int i = 0; i < arr.length; i++) {
            int newIndex = i + offset;
            if(newIndex >= arr.length) {
                  newIndex -= arr.length;
            } else if (newIndex < 0) {
                  newIndex = arr.length + newIndex;
            }
            shifted[newIndex] = arr[i];
      }
      return shifted;
}

// v2

public static int[] shift(int[] arr, int offset) {
      int[] shifted = new int[arr.length];
      int newIndex;
      if (Math.abs(offset) >= arr.length) {
            offset = offset % arr.length;
      }
      for(int i = 0; i < arr.length; i++) {
            newIndex = i + offset;
            if (newIndex >= arr.length) {
                  newIndex = newIndex % arr.length;
            } else if (newIndex < 0) {
                  newIndex = arr.length + newIndex % arr.length;
            }
            shifted[newIndex] = arr[i];
      }
      return shifted;
}
// or two sequential loops
// don't use strings, though!
```

11. Draw the output of the code on the left in the grid on the right. If you need to write a space character, just leave a box blank. You do not have to use all of the squares. (3 points)

```
int size = 4;
for (int i = size; i > 0; i--) {

  for (int j = i ; j < size; j++) {
    // print out two spaces
    System.out.printf(" ");
  }

  for (int j = 1; j <= i; j++) {
    // space only
    System.out.printf(" %s", j);
  }

  // just for the new line!
  System.out.println();
}
```

	1		2		3		4		
			1		2		3		
					1		2		
							1		

12. Answer the following questions about characters, ints, binary and hexadecimal. (3 points)

(a) What is the output of the following code? See reference material on last page for character codes.

```
char ch = '\u0045';
syso(ch);
```

E

```
int i = 0b00001101;
syso(i);
```

13

(b) What is 176 (decimal, base 10) in hexadecimal? What is 3 (base 10, decimal) in binary?

B0

11

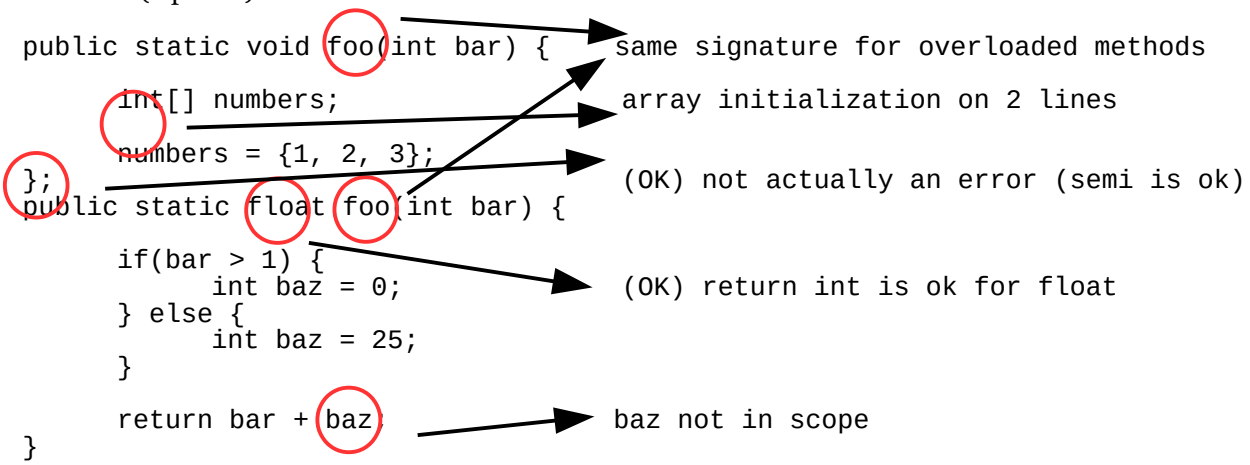
13. Using the program below, draw the stack after each method call. (3 points)

```
public static void main(String[] args) {
  foo();
}
public static void foo() {
  int i = 5;
  int j = bar(i);
  System.out.println(j);
}
public static int bar(int baz) {
  int result = baz * 2;
  return result;
}



// view in pythontutor (http://www.pythontutor.com/)Java mode to see
// interactive progression of call stack
```

14. **Circle 4 errors** in this code and give a short (2 or 3 word) description of the error. Draw arrows to help annotate. (2 points)



15. Short answer questions. **Choose any three** to answer. Every questions answered beyond 3 will count as extra credit. (5 points total – 3 regular, 2 extra credit)

(a) What is type casting? When do we need to explicitly type cast?

Type casting is converting from one type to another - (int) 1.2. We need to explicitly type cast when we're performing a narrowing conversion.

(b) In the context of type casting, what is meant by widening a type and narrowing a type?

widening – going from a smaller range to a larger range
narrowing – going from a larger range to a smaller range

(c) Why does floating point arithmetic yield results that look slightly inaccurate? Describe one way to mitigate floating point arithmetic issues.

Some numbers are not accurately representable using binary. This can be mitigated by using doubles, by not relying on floating point calculations for counting conditions and by performing repeated floating point calculations from smallest to largest.

(d) Name the two broad categories of types... and give an example of a type in each category.

primitives – int, reference types – any Array (int[])

(e) Why must you be careful when passing an Array as an argument to a method? What is *actually* being passed?

The value of the reference is being passed... so there may be side effects when working with the arguments that are passed in.