CSAP 1st Quarterly Review - List of Topics

What is Java?

A fast growing object-oriented programming language. It is secure, robust and portable.

Evaluating arithmetic expressions with mixed data types, operator precedence, integer division

```
Ex 1:  5 * 3 – ( 4 + 6 * 3)  / 4                    //10
Ex 2:  38 % 4 + (12 + 4 * 3) / 5.0                  //6.8
```

Assignment statements with mixed data types and type casting

```
Ex 3: int myVar = 3.0 * 4;                          //error
Ex 4: double myVar = (int) (3.0 * 4.1);             //12.0
Ex 5: double myVar = (double )  3 / 4;              //0.75
Ex 6: double myVar = (double) (3/4);                //0.0
```

Concatenation vs addition

```
Ex 7:  System.out.print ( "Sum of 1 + 2 + 3 is " + 1 + 2 + 3);
                                                    //Sum of 1 + 2 + 3 is 123
Ex 8:  System.out.print (1 + 2 + 3 + " is the sum of 1 + 2 + 3");
                                                    //6 is the sum of 1 + 2 + 3
Ex 9:  System.out.print (1 + 2 + 3 + " is the sum of " + (1 + 2 + 3));
                                                    //6 is the sum of 6
```

Math methods: sqrt, pow, min, max

```
Ex 10:  Math.sqrt ( Math.pow (3,2) )                //3.0
Ex 11:  Math.pow (Math.sqrt(36),26%3)               //36.0
```

Types of programming errors

1. Syntax or Compile
2. Runtime
3. Logic

Types of errors in loops

1. OBOE
2. Infinite
3. Update
4. Initialization

deMorgan's Law (pg. 235)

```
Ex 11:  !(p && q)                   // !p || !q
Ex 12:  !(p || q)                   // !p && !q
Ex 13:  p && (q || r)               // (p && q) || (p && r)
Ex 14:  p || ( q && r)              // (p || q) && (p || r)
```

Escape sequences

```
\t    \n    \"     \\
```

Comments

```
//        /*     */      /**     */
```

Addition vs. Concatenation

Sample Open-ended

Assume 3 `int` variables `a`, `b` and `c` have been declared and initialized. Write a Java code segment that prints the product of `b` and `c` when `a` is greater or equal to zero and the sum of `b` and `c` when `a` is negative.

```
if (a >= 0)
      System.out.print (“product of b and c” + (b * c));
else
      System.out.print (“sum of b and c is “ + (b + c));
```