COE318 Midterm Study Guide

The study guide covers many of the topics you should understand in preparation for the midterm. However, it is not comprehensive and should be viewed as minimal preparation. In particular, the midterm will not contain any of these questions and could contain questions covering topics not included in the study guide.

How to use the guide

The guide consists of questions and answers. Each question also indicates the estimated maximum time it should take you to answer the question. It is recommended that you try to answer the question before looking at the answer.

1. (10 *minutes)* Consider the following class:

```
public class F {

    public static boolean f(int a, int b) {
        return (a > b ? a : b) % 3 == 0;
    }

    public static void main(String[] args) {
        for(int i = 1; i < 8; i++) {
            System.out.println("f(" + i + ", 3): " + f(i, 3));
        }
    }
}
```

Answer:

```
f(1, 3): true
f(2, 3): true
f(3, 3): true
f(4, 3): false
f(5, 3): false
f(6, 3): true
f(7, 3): false
```

2. (*10 minutes)* What is the output when the following is run?

```
public class F {
```

```
    public static void main(String[] args) {
        int k = 0;
        for(int i = 0; i < 10; i += 2) {
            for(int j = 1; j < 16; j *= 2) {
                if(i >= 2 && i <= 4) {
                    k++;
                }
            }
        }
        System.out.println("k: " + k);
    }
}
```

Answer:

```
k: 8
```

3. (*10 minutes)* What is the output when the following is run?

```
public class F {
    public static void main(String[] args) {
        k = 0;
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 5; j++) {
                if (j % 2 == 0) {
                    continue;
                } else if (j == 3) {
                    break;
                }
                k++;
            }
        }
        System.out.println("k: " + k);
    }
}
```

Answer:

```
k: 10
```

4. (*5 minutes*) What is the output from the following?

```
      public static void main(String[] args) {
          String s1 = "coe31";
          s1 += "8";
          String s2 = "coe318";
          String s3 = s1;
```

```
        System.out.println(s1 == s2);
        System.out.println(s1 == s3);
        System.out.println(s2 == s3);
        System.out.println(s1.equals(s2));
        System.out.println(s1.equals(s3));
        System.out.println(s2.equals(s3));
    }
```

Answer:

```
false
true
false
true
true
true
```

5. (*5 minutes*) What is the output from the following?

```
public class F {
  public static void main(String[] args) {
     System.out.println((3>=5 || 5<6) + "1" + (1 1) + "\"1\"");
     System.out.println("" + 1 + 1 + (2 * 3 + 1) + "(5 * 7)");
  }
}
```

Answer:

```
true12"1"
117(5 * 7)
```

6. (*5 minutes*) What is the output from the following?

```
public class F {
    public static int g(int a) {
        int s = 0;
        switch(a) {
            case 0:
                s++;
            case 1:
                s++;
                break;
            case 2:
                s++;
            case 3:
                s--;
            case 4:
                s++;
                break;
            default:
                s = -5;
        }
        return s;
```

```
    }

    public static void main(String[] args) {
        for(int i = 0; i < 6; i++) {
            System.out.println("g(" + i + "): " + g(i));
        }
    }
}
```

Answer:

```
g(0): 2
g(1): 1
g(2): 1
g(3): 0
g(4): 1
g(5): -5
```

7. (2*0 minutes)* What is the output when the following is run?

```
package coe318;

public class F {

    public static boolean f(int a, int b) {
        return (a > b ? a : b) % 3 == 0;
    }

    public static int g(int a) {
        int s = 0;
        switch (a) {
            case 0:
                s++;
            case 1:
                s++;
                break;
            case 2:
                s++;
            case 3:
                s--;
            case 4:
                s++;
                break;
            default:
                s = (a < 0) ? (g(++a) + 1) : (g(--a) - 1);
        }
        return s;
    }

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

```
        for (int i = -3; i < 9; i++) {
            System.out.println("g(" + i + "): " + g(i));
        }
    }
}
```

Answer:

```
g(-3): 5
g(-2): 4
g(-1): 3
g(0): 2
g(1): 1
g(2): 1
g(3): 0
g(4): 1
g(5): 0
g(6): -1
g(7): -2
g(8): -3
```

8. In the following class each method (t1, t2, t3, t4 and t5) contains **exactly** one error. Each method has an array of ints as its argument and it is supposed to return the sum of all of the array elements. The error may be one of the following:
   - an error that the compiler will detect (i.e. it will not compile)
   - something that will compile but produce a *runtime* error, crashing the program.
   - A logic error; the code will compile and run but the answer will be wrong.

   Identify and fix each error. (Identify whether it is compile, runtime or logic error.)

```
public class Total {

    public static int t1(int[] a) {
        int total = 0;
        for(int i = 1.0; i <= a.length; i++) {
            total = total + a[i - 1];
        }
        return total;
    }

    public static int t2(int[] a) {
        int total = 0;
        for (int i = 0; i <= a.length; i++) {
            total = total + a[i];
        }
        return total;
    }
```

```
    public static int t3(int[] a) {
        int t;
        for (int i : a) t += i;
        return t;
    }

    public static int t4(int[] a) {
        int t;
        for (int i = 1; i <= a.length; i++) {
            t += a[i - 1];
        }
        return t;
    }

     public static int t5(int[] a) {
        int t = 0;
        for (int i = 1; i < a.length; i++) {
            t += a[i];
        }
        return t;
    }

    public static void main(String[] args) {
        int[] nums = {1, 2, 3, 4, 5};
        System.out.println("t1: " + t1(nums));
        System.out.println("t2: " + t2(nums));
        System.out.println("t3: " + t3(nums));
        System.out.println("t4: " + t4(nums));
        System.out.println("t5: " + t5(nums));
    }
}
```

Answer:

```
public class Total {

    public static int t1(int[] a) {
        int total = 0;
//Compile time error (use the int 1 or cast (int) 1.0
        for(int i = 1; i <= a.length; i++) {
        // for(int i = 1.0; i <= a.length; i++) {
            total = total + a[i - 1];
        }
        return total;
    }
```

```
    public static int t2(int[] a) {
        int total = 0;
//runtime error (array index out of bounds)
        for (int i = 0; i < a.length; i++) {
       // for (int i = 0; i <= a.length; i++) {
            total = total + a[i];
        }
        return total;
    }

    public static int t3(int[] a) {
//Compile error (use of uninitialized local variable)
      //  int t;
        int t = 0;
        for (int i : a) t += i;
        return t;
    }

    public static int t4(int[] a) {
      //Compile error (use of uninitialized local variable)
        //int t;
        int t = 0;
        for (int i = 1; i <= a.length; i++) {
            t += a[i - 1];
        }
        return t;
    }

     public static int t5(int[] a) {
        int t = 0;
//Logic error (does not include a[0]
        //for (int i = 1; i < a.length; i++) {
        for (int i = 0; i < a.length; i++) {
            t += a[i];
        }
        return t;
    }

    public static void main(String[] args) {
        int[] nums = {1, 2, 3, 4, 5};
        System.out.println("t1: " + t1(nums));
        System.out.println("t2: " + t2(nums));
        System.out.println("t3: " + t3(nums));
        System.out.println("t4: " + t4(nums));
        System.out.println("t5: " + t5(nums));
    }
}
```