COSC 117 **Exam 2 Key** Fall 2012

Part 1: Definitions & Short Answer (3 Points Each)

1. What Preconditions and Postconditions?

   A precondition of a subroutine is something that must be true when the subroutine is called and a postcondition of a subroutine is something that will be true after the subroutine has run

2. What are parameters?

   Parameters are part of the interface of a function which allows values to be passed from the call to the function.

3. What is the difference between formal parameters and actual parameters?

   Parameters in a subroutine definition are called formal parameters or dummy parameters. The parameters that are passed to a subroutine when it is called are called actual parameters or arguments.

4. What is a return type and where is it designated in the function header?

   The return type is the type of value that is returned by the function, if a value is to be returned. The position of return type is after the modifiers and directly before the name of the function. For example, in the following header the return type is int.

   ```
   public static int doSomething(int x, int y)
   ```

5. What is the scope of a function parameter?

   The scope of a function parameter is just in the function itself and not outside of the function.

Part 2: Program Traces (15 Points Each)

1. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class Exam2Trace1 {

   public static int doSomething(int x, int y) {
      while (x != y) {
         if (x > y)
            x -= y;
         else
            y -= x;
      }
      return x;
   }

   public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Input x: ");
      int x = keyboard.nextInt();
      System.out.print("Input y: ");
      int y = keyboard.nextInt();

      System.out.println("Result = " + doSomething(x, y));
   }
}
```

(a)

```
Input x: 7
Input y: 9

Result = 1
```

(b)

```
Input x: 25
Input y: 15

Result = 5
```

(c)

```
Input x: -2
Input y: 8
```

Infinite loop, no output.

2. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;
public class Exam2Trace2 {

    public static void doSomethingMore(int x, int y, int z) {
        z *= y;
        x = -x + z;
        y = 3 * y - 2;
        System.out.println(x + "  " + y + "  " + z);
    }

    public static void doSomething(int x, int y, int z) {
        x += y;
        z = 3 * x;
        y = Math.max(x, y);
        System.out.println(x + "  " + y + "  " + z);
        doSomethingMore(z, y, x);
        System.out.println(x + "  " + y + "  " + z);
    }

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Input x: ");
        int x = keyboard.nextInt();
        System.out.print("Input y: ");
        int y = keyboard.nextInt();
        System.out.print("Input z: ");
        int z = keyboard.nextInt();

        doSomething(y, x, z);
        doSomethingMore(z, x, y);
        System.out.println(x + "  " + y + "  " + z);
    }
}
```

(a)

```
Input x: 1
Input y: 2
Input z: 3

3  3  9
0  7  9
3  3  9
-1  1  2
1  2  3
```

(b)

```
Input x: 3
Input y: 5
Input z: 2

8  8  24
40  22  64
8  8  24
13  7  15
3  5  2
```

(c)

```
Input x: -3
Input y: 2
Input z: 5

-1  -1  -3
4  -5  1
-1  -1  -3
-11  -11  -6
-3  2  5
```

3. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class Exam2Trace3 {

   public static int alter(int x, int y) {
      int t = 0;
      if (y > 20)
         t = --y;
      else
         t = y++;
      return ++t;
   }

   public static boolean isit(int x, int y) {
      return (3*x < 2*y);
   }

   public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Input x: ");
      int x = keyboard.nextInt();
      System.out.print("Input y: ");
      int y = keyboard.nextInt();

      while (isit(x, y)){
         System.out.println(x + "  " + y);
         x = alter(--y, x);
      }
      System.out.println(x + "  " + y);
   }
}
```

(a)

```
Input x: 4
Input y: 15

4  15
5  14
6  13
7  12
8  11
```

(b)

```
Input x: 2
Input y: 3

2  3
```

(c)

```
Input x: -3
Input y: 4

-3  4
-2  3
-1  2
0  1
1  0
```

Part 3: Coding (10 Points Each)

1. Write a function that rolls two die and returns the sum of the face values. Also write a call for the function that stores the result in a variable called roll.

```
public static int rollDie() {
    int die1 = (int) (Math.random() * 6) + 1;
    int die2 = (int) (Math.random() * 6) + 1;
    return die1 + die2;
}

int roll = rollDie();
```

2. Write a function that takes as input (as a parameter) an average for a student in a class on the 0 – 100 scale and returns the student's letter grade as a character. The grade should be determined on the 90-80-70-60 scale. The function is to return the letter grade to the calling function and not print it out. Also write a call for the function that stores the result in a variable called grade.

```
public static char letterGrade(double avg) {
   if (avg >= 90)
       return 'A';
   else if (avg >= 80)
       return 'B';
   else if (avg >= 70)
       return 'C';
   else if (avg >= 60)
       return 'D';
   else
       return 'F';
}

char grade = letterGrade(87.6);
```

3. Write a function called getInteger1_10 that will continually ask the user for an integer in the range from 1 to 10 until the user inputs an integer in this range. The function should then return the input number to the call. The function must also handle any run-time error from user input. Also write a call for the function that stores the input number in a variable called x.

```
public static int getInteger1_10(){
    int input = 0;
    Scanner keyboard = new Scanner(System.in);

    while (input < 1 || input > 10){
       try{
           System.out.print("Input an integer between 1 and 10: ");
           input = keyboard.nextInt();
           if (input < 1 || input > 10)
              System.out.println("Input is out of range, please try again.");
       } catch (Exception e)
       {
           System.out.println("Invalid Input, please try again.");
           String dumb = keyboard.nextLine();
       }
    }
    return input;
}

int x = getInteger1_10();
```

4. Write a function that takes as input (as a parameter) a string and returns the number of vowels in the string. For this function the vowels do not include y. Also write a call for the function that stores the result in a variable called v.

```
public static int getNumVowels(String str){
    str = str.toLowerCase();
    int count = 0;
    for (int i = 0; i < str.length(); i++){
        char ch = str.charAt(i);
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
            count++;
    }
    return count;
}

int v = getNumVowels("This is a test of the VOWEL COUNTER");
```

5. Write a function called Area that will take as input (as parameters) the lengths of the three sides of a triangle and return the area of the triangle. Recall that given the lengths of the three sides of a triangle, a, b, and c the area can be found by $\sqrt{s(s-a)(s-b)(s-c)}$ where s is the semi-perimeter given by, $\frac{a+b+c}{2}$. Also write a call for the function assuming that the function is contained in a class called Triangle, store the result in a variable named area.

```
public static double Area(double a, double b, double c) {
    double s = (a+b+c)/2;
    return Math.sqrt(s*(s-a)*(s-b)*(s-c));
}

double area = Triangle.Area(a,b,c)
```