Fall 2013 **Exam #2 Key** COSC 117

# 1 Short Answer (2 Points Each)

1. What is the scope of a method/function parameter?

   **Solution:** The scope of a method/function parameter is in the method only, that is, it is local to the method and cannot be seen outside the method.

2. What is method/function overloading? How is it done in Java?

   **Solution:** Method overloading is when two methods have the same name, in Java two methods can share the same name as long as their parameter lists are different.

3. Give an example of how you can detect if there is an integer waiting to be read from the input buffer.

   **Solution:**

```
Scanner keyboard = new Scanner(System.in);
int value;

if (keyboard.hasNextInt()) {
    value = keyboard.nextInt();
} else
    System.out.print("ERROR");
```

4. Write the Java code around the line `i = j/k;` that will keep the program from crashing if the value of `k` is zero and if the value of `k` is zero it should set the value of `i` to 0. You may not use any `if` or `switch` statements to do this.

   **Solution:**

```
try{
    i = j/k;
} catch(Exception e) {
    i = 0;
}
```

5. What is wrong with the following program and how would you fix the error? If there is no error simply state that there is no error.

```
1  public class Exam2SA2 {
2
3      public static int DoThis(boolean b, int i, String str){
4          int k = 0;
5          if (b){
6              k = i + str.indexOf("a");
7          }
8          return k;
9      }
10
11     public static void main(String[] args) {
12         int i = -4;
13         String str = "Exam 2";
14         boolean b = true;
15         System.out.println(DoThis(str, b, i));
16     }
17 }
```

   **Solution:** The error is in the order of the parameters in the function call. A correction would be

   `System.out.println(DoThis(b, i, str));`

## 2 Program Traces (15 Points Each)

1. Consider the following program.

```
1  import java.util.Scanner;
2
3  public class Exam2Trace1 {
4
5      public static int DoThis(int a, int b){
6          while(b != 0){
7              int t = a;
8              a = b;
9              b = t % b;
10             System.out.print(a + "  ");
11         }
12         return a;
13     }
14
15     public static void main(String[] args) {
16         Scanner keyboard = new Scanner(System.in);
17         System.out.print("Input x: ");
18         int x = keyboard.nextInt();
19         System.out.print("Input y: ");
20         int y = keyboard.nextInt();
21         System.out.println(DoThis(x,y));
22     }
23 }
```

   (a) Write the output of the program for the given input.

```
Input x: 12
Input y: 8

8  4  4
```

   (b) Write the output of the program for the given input.

```
Input x: 36
Input y: 15

15  6  3  3
```

   (c) Write the output of the program for the given input.

```
Input x: 36
Input y: 100

100  36  28  8  4  4
```

   (d) What does the method `DoThis` calculate?
   **Solution:** `DoThis` calculate the greatest common divisor of the two input numbers.

2. For each of the given inputs, write the output of the program.

```
1  import java.util.Scanner;
2
3  public class Exam2Trace2 {
4
5      public static int DoThis(int a, int b){
6          int k = 0;
7          int j = 0;
8
9          k = a/b;
10         if (k % 3 == 0)
11             j = 2*k;
12         else if (k % 2 == 0)
13             j = 3*(k++);
14         else
15             j = 5;
16
17         k = (int)(j * 1.5);
18         System.out.print(a + "  " + b + "  " + k + "  " + j + "  ");
19         return k;
20     }
21
22     public static void main(String[] args) {
23         Scanner keyboard = new Scanner(System.in);
24         System.out.print("Input x: ");
25         int x = keyboard.nextInt();
26         System.out.print("Input y: ");
27         int y = keyboard.nextInt();
28         System.out.println(DoThis(x,y));
29     }
30 }
```

(a)
```
Input x: 23
Input y: 7

23  7  9  6  9
```

(b)
```
Input x: 149
Input y: 11

149  11  7  5  7
```

(c)
```
Input x: 149
Input y: 10

149  10  63  42  63
```

3. Consider the following program.

```
1  import java.util.Scanner;
2
3  public class Exam2Trace3 {
4
5      public static int DoEvenMore(int a, int b){
6          if (a > b){
7              int temp = a;
8              a = b;
9              b = temp;
10         }
11         return b;
12     }
13
14     public static int DoMore(int a, int b){
15         if (a > b){
16             int temp = a;
17             a = b;
18             b = temp;
19         }
20         return a;
21     }
22
23     public static String DoThis(int x, int y, String str
           ){
24         int k = str.indexOf("a");
25         int j = str.lastIndexOf("e");
26         int a = 5;
27         int b = 7;
28
29         if (k >= 0 && j >= 0){
30             if (k > j){
31                 int temp = k;
32                 k = j;
33                 j = temp;
34             }
35             str = str.substring(k, j);
36         } else{
37             a = DoMore(x, y);
38             b = DoEvenMore(x, y);
39             str = str.substring(a, b);
40         }
41
42         System.out.println(k + "  " + j + "  " + x + "
               " + y + "  " + a + "  " + b + "  ");
43         return str;
44     }
45
46     public static void main(String[] args) {
47         Scanner keyboard = new Scanner(System.in);
48         System.out.print("Input str: ");
49         String str = keyboard.nextLine();
50         System.out.print("Input x: ");
51         int x = keyboard.nextInt();
52         System.out.print("Input y: ");
53         int y = keyboard.nextInt();
54
55         System.out.println(DoThis(x,y,str));
56     }
57 }
```

(a) Write the output of the program for the given input.

```
Input str: This is a test
Input x: 3
Input y: 8

8  11  3  8  5  7
a t
```

(b) Write the output of the program for the given input.

```
Input str: Be Nice
Input x: 2
Input y: 7

-1  6  2  7  2  7
 Nice
```

(c) Write the output of the program for the given input.

```
Input str: This string is long
Input x: 16
Input y: 5

-1  -1  16  5  5  16
string is l
```

(d) What do the methods `DoMore` and `DoEvenMore` calculate? Write a one line function body for each.

**Solution:** `DoMore` returns the minimum of $a$ and $b$ and `DoEvenMore` returns the maximum of $a$ and $b$. So a one line function body for these would be

```
public static int DoMore(int a, int b){
    return Math.min(a, b);
}
```

and

```
public static int DoEvenMore(int a, int b){
    return Math.max(a, b);
}
```

respectively.

# 3 Coding

1. (*10 Points*) Write a method called `Roll2Die` that will simulate the roll of two dice. The method should return 1 if the roll was "snake eyes" (that is two ones), a 2 if the roll was "box cars" (that is two 6's) and it should return 0 if the roll was neither. The method should use a switch statement to determine the output. Also, write a call to this function that will store the result in a variable named `roll`, write just the function call not the entire main program.

**Solution:** Code:

```
public static int Roll2Die() {
    int die1 = (int)(Math.random()*6+1);
    int die2 = (int)(Math.random()*6+1);
    int sum = die1 + die2;
    int retval = 0;

    switch (sum){
    case 2:
        retval = 1;
        break;
    case 12:
        retval = 2;
        break;
    default:
        retval = 0;
    }

    return retval;
}
```

Function Call:

```
int roll = Roll2Die();
```

2. (*15 Points*) Write a method called `getInteger` that will take two integer input parameters called `low` and `high`, have it get an integer value from the user that is between these two numbers and return that value. If the user types in an integer that is outside of the given range the method should return the one less than the value of `low` as an error code. If the user types in anything other than an integer the method should return one more than the value of `high` as an error code. Also, write a call to this function that will get a number between 100 and 150 and store the input number, or the error code, in a variable named `num`, write just the function call not the entire main program. You do not have to worry about a fancy prompt for the user, `Input:` will be fine. Also do not worry about printing out any errors to the screen.

**Solution:** Code:

```
public static int getInteger(int low, int high){
    Scanner keyboard = new Scanner(System.in);
    int value;
    System.out.print("Input: ");
    try{
        value = keyboard.nextInt();
        if(value < low || value > high)
            value = low-1;
    } catch(Exception e) {
        value = high + 1;
    }
    return value;
}
```

Function Call:

```
int num = getInteger(100, 150);
```

3. (*15 Points*) Write a method called `IncomeTax` that will take as input a decimal number representing the user's yearly taxable income and return the amount of income tax they must pay the government. Income tax is calculated as a percentage of the user's income according to what bracket they fall in. Here is the tax scheme. If the person makes less than $40,000 they pay 20% of their income in tax. If the person makes $40,000 or more up to but not including $60,000 they pay $8,000 plus 25% of their income that exceeds $40,000 in tax, that is, if they make $50,000 they pay $8,000 plus 25% of $10,000. If the person makes $60,000 or more up to but not including $100,000 they pay $13,000 plus 27.5% of their income that exceeds $60,000 in tax. If the person makes $100,000 or more they pay $24,000 plus 30% of their income that exceeds $100,000 in tax.

   **Solution:**

```
public static double IncomeTax(double income){
    double tax = 0;

    if (income < 40000){
        tax = income * 0.2;
    } else if (income < 60000){
        tax = 8000 + 0.25*(income - 40000);
    } else if (income < 100000){
        tax = 13000 + 0.275*(income - 60000);
    } else {
        tax = 24000 + 0.3*(income - 100000);
    }

    return tax;
}
```

4. (*15 Points*) Write a method called `BlackJack` that does not take in any parameters but returns an integer that designates the winner of a single hand of Black Jack. This is a simplified version of Black Jack, in this game two players are dealt two cards and the winner is determined from that deal. That is, there is no user interaction for being dealt any more cards. The method should deal two cards at random to each of two players, do not worry about duplicate cards. In this game, Ace is worth 11 points, face cards are all 10 and all other cards are their face value. A "Black Jack" is when a person is dealt and Ace and Jack and it is the highest possible hand which wins over any other hand. If the person does not have a Black Jack then the worth of their hand is the sum of the card values, unless the player has two aces in which case each ace is converted to being value 1 and hence the hand is worth 2. The method returns the winner of the game (1 for player #1 and 2 for player #2). The method should return 0 if the game is a draw.

   **Solution:**

```
public static int BlackJack(){
    int Player1Card1 = (int)(Math.random()*13+1);
    int Player1Card2 = (int)(Math.random()*13+1);
    int Player2Card1 = (int)(Math.random()*13+1);
    int Player2Card2 = (int)(Math.random()*13+1);

    int Player1Value = 0;
    if (Player1Card1 == 1)
        Player1Value = 11;
    else if (Player1Card1 >= 10)
        Player1Value = 10;
    else
        Player1Value = Player1Card1;

    if (Player1Card2 == 1)
        Player1Value += 11;
    else if (Player1Card2 >= 10)
        Player1Value += 10;
    else
        Player1Value += Player1Card2;

    if (Player1Value == 22)
        Player1Value = 2;
    else if ((Player1Value == 21) && (Player1Card1 == 11
          || Player1Card2 == 11))
        Player1Value = 100;

    int Player2Value = 0;
    if (Player2Card1 == 1)
        Player2Value = 11;
    else if (Player2Card1 >= 10)
        Player2Value = 10;
    else
        Player2Value = Player2Card1;

    if (Player2Card2 == 1)
        Player2Value += 11;
    else if (Player2Card2 >= 10)
        Player2Value += 10;
    else
        Player2Value += Player2Card2;

    if (Player2Value == 22)
        Player2Value = 2;
    else if ((Player2Value == 21) && (Player2Card1 == 11
          || Player2Card2 == 11))
        Player2Value = 100;

    if (Player1Value > Player2Value)
        return 1;
    else if (Player1Value < Player2Value)
        return 2;
    else
        return 0;
}
```