

## CS125 Spring 2009 Exam 3 Review Key

Before you start, please note: it is possible to get strong clues for answering some of the questions in the review by referring to other questions. For example, converting from code to a flow chart and vice versa. In general, you should not depend on both versions of a question showing up in the exam.

Also note: this study guide is not comprehensive. I strongly suggest reviewing: the practice problems posted online, homework assignments, material from the last exam & the self check questions in the text.

Write a public method called `convertToArray` that takes an `ArrayList` of type `Integer` as a parameter and returns a corresponding array of type `int`.

// code goes here

```
/** converts the given array to an ArrayList
 * @param someList the ArrayList to convert
 * @return the corresponding array
 */
public int[] convertToArray(ArrayList<Integer> someList)
{
    int[] array = new int[someList.size()];
    for(int i =0; i<array.length; i++)
    {
        array[i] = someList.get(i);
    }
    return array;
}
```

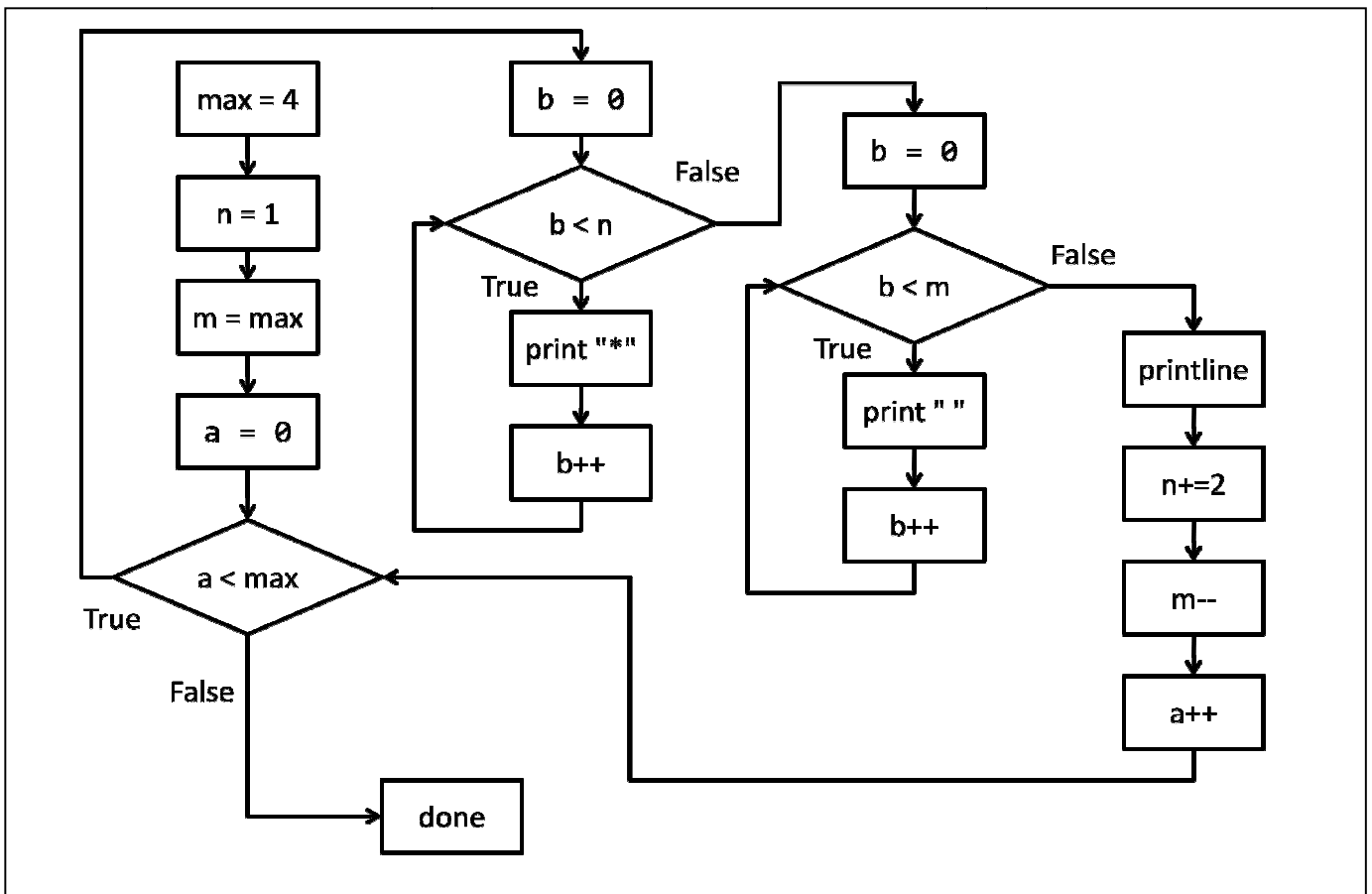
Examine the following code:

```
int max = 4;
int n = 1;
int m = max;
for(int a=0; a<max; a++)
{
    for(int b=0; b<m; b++)
        System.out.print(" ");
    for(int b=0; b<n; b++)
        System.out.print("*");
    System.out.println("");
    n += 2;
    m--;
}
```

final output is:  
\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

What is the final output displayed by the code:

Diagram the code using a flow chart:



Examine the code below. **Add** the necessary lines of code such that each new Player object will get its own unique idNum. Note: this should not be done by changing any of the existing code.

```
public class Player
{
```

```
    // instance fields
    int score;
    int idNum;
```

```
private static int lastIdNum = 0;
```

```
/**
 * constructor - sets idNum
 */
public Player()
{
```

```
    idNum = lastIdNum;
    lastIdNum++;
```

```
}
```

```
/**
 * returns idNum
 * @return the objects idNum
 */
public int getIdNum()
{
    return idNum;
}
```

```
}
```

Write a for each loop that sums up the contents for an ArrayList called values, stores the result in double called sum & then prints it.

// code goes here

```
double sum = 0;
for(Double v : values)
{
    sum += v;
}
System.out.println(sum);
```

Write a for loop that finds the largest value in an array called values, and swaps it to the last position of the array

// code goes here

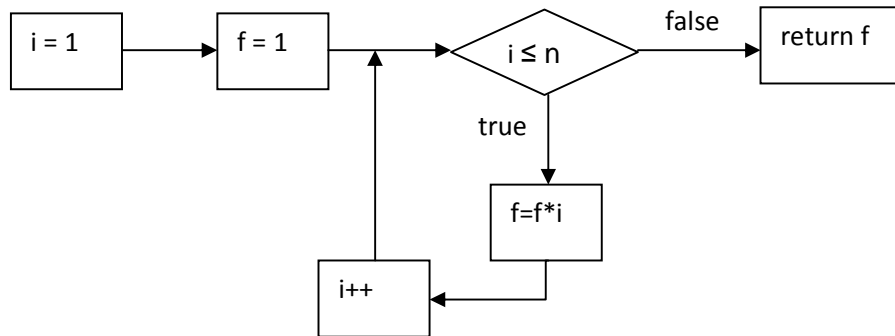
```
// find index of largest element
int maxPosition = 0;
for(int i=1; i<values.length; i++)
{
    if(values[i] > values[maxPosition])
        maxPosition = i;
}
// perform the swap
int lastPosition = values.length-1;
double temp = values[lastPosition];
values[lastPosition] = values[maxPosition];
values[maxPosition] = temp;
```

Write a while loop that examines all the values in an integer array called values, finds the number of the values that are odd, stores the result in a variable called oddCount & then prints it.

// code goes here

```
int i=0;
int oddCount=0;
while(i < array.length)
{
    if(array[i] %2 == 0 && array[i] != 0)
        count++;
}
System.out.println(oddCount);
```

Examine the following flow chart & convert into a static method called `factorial` which takes a `int` parameter `n` and returns an `int` as an answer. Be sure to include documentation comments for the method.



// code goes here

note: `int` will work in the code below, but `long` would be better – factorials quickly yield very large numbers!

```

public static int factorial(int n)
{
    i=1;
    f=1;
    while(i <= n)
    {
        f = f*i;
        i++;
    }
    return f;
}
  
```

Define coupling.

Coupling is the degree to which classes are dependent upon each other. High coupling results in code that is more complicated, difficult to maintain and error prone.

What is a postcondition? Give an example.

A precondition is a situation that must be true for a method to work correctly. It is the responsibility of the calling code to make sure this is the case. Example: if you use `Math.sqrt(x)`, you must ensure `x ≥ 0`

What does parameter recycling mean? Give an example.

Parameter recycling occurs when a method reuses a parameter or local variable, in an inappropriate way for something else (example on next page)

```

public static double calcSphereSurfaceArea(double radius)
{
    radius = 4.0 * Math.PI * radius * radius;
    return radius;
}

```

radius is being recycled to hold surface area calculation

Write a class called `MoreMath` with a static method called `GreatestCommonDenominator`. It should take two integers as parameters & return their greatest common denominator – that is, it should find the largest number that divides evenly into both numbers. Provide documentation comments including a precondition comment indicating the method does not take negative numbers.

```

/**
 * holds additional math routines
 */
public class MoreMath
{
    /** deteremines the GCD of the given numbers
     *
     * @param x    the first number to examine
     * @param y    the second number to examine
     * @precondition x >= 0
     * @precondition y >= 0
     * @return the largest int g such that x % g == 0 && y % g == 0
     */
    public static int GreatestCommonDenominator(int x, int y)
    {
        // if nothing else, both must divide by 1
        int gcd = 1;
        for(int a = 2; a < x && a < y; a++)
        {
            if( ((x % a) == 0) && ((y % a) == 0) )
                gcd = a;
        }

        return gcd;
    }
}

```