

## CS125 Fall 2008 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.

Fill in some code to print out whether or not a number is positive.

For example:

if the user enters 6, display: "6 is positive"

if the user enters -3, display: "-3 is negative"

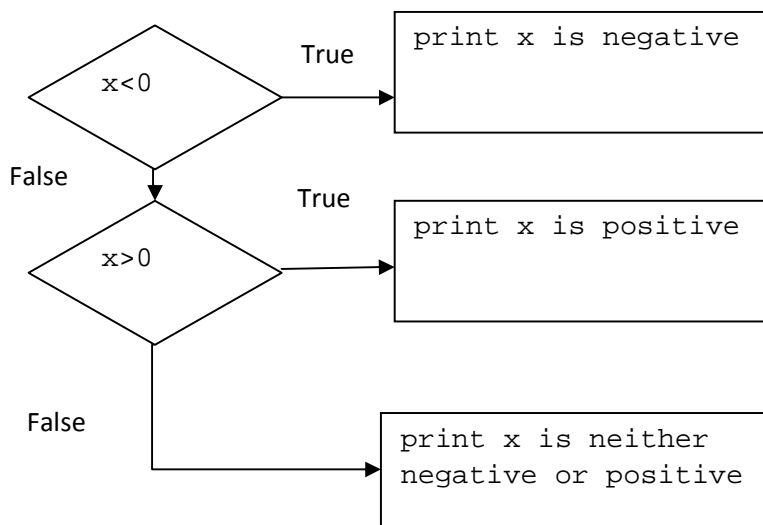
if the user enters 0, display: "0 is neither positive or negative"

```
// get input from user & save it in x
int x = in.nextInt();
```

```
// your code goes below here
```

```
if(x > 0)
    System.out.println(x + " is positive");
else if(x < 0)
    System.out.println(x + " is negative");
else
    System.out.println(x + " is neither positive or negative");
```

Write out a flow chart for the code you wrote above:



Add in a single line of code to test if  $-5 < x < 0$  or if  $10 < x < 20$

```
// get input from user & save it in x
int x = in.nextInt();
```

```
// your code goes below here
```

```
if((-5 < x && x < 0) || (10 < x && x < 20))
```

```
// your code goes above here
```

```
{
    System.out.println("value is in the range -5 to 0 or 10 to 20);
}
```

Write a public method called `convertToArrayList` that takes an array of integers as a parameter and returns a corresponding `ArrayList`.

```
// code goes here
```

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

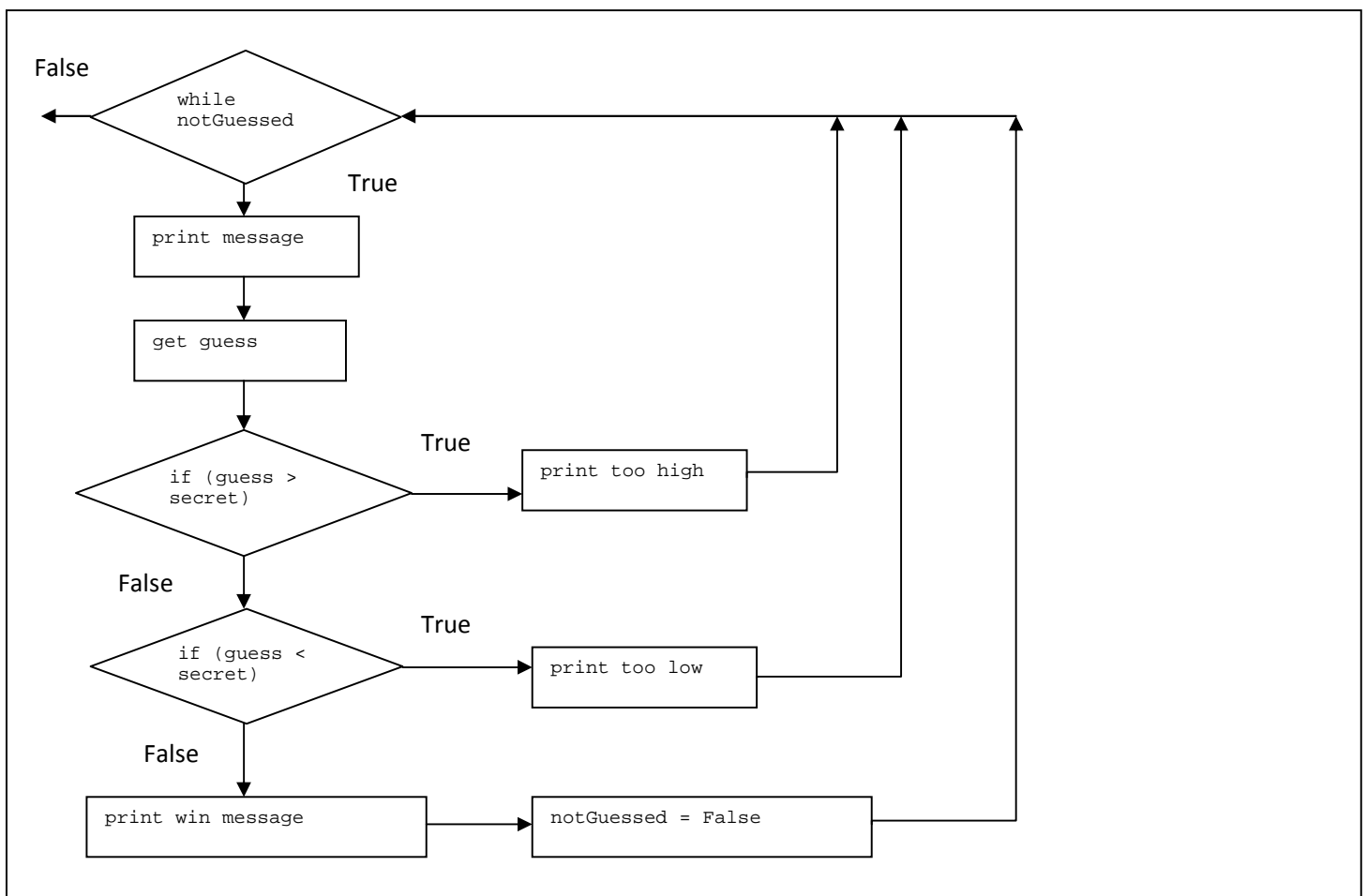
Write out the flow chart for the following code:

```
// get random number
int secret = randGen.nextInt(11);
boolean guessedCorrect = false;
int numGuesses = 1;

// loop until guessed correctly
while(!guessedCorrect)
{
    // get input
    System.out.println("Enter an integer between 0 & 10: ");
    int guess = in.nextInt();

    // print response
    if(guess > secret)
        System.out.println("too high");
    else if(guess < secret)
        System.out.println("too low");
    else
        System.out.println("you won in " + numGuesses + " tries!");
    numGuesses++;
}
```

Flow chart goes below here.



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 array 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 `ArrayList` called `values`, stores the result in double called `max` & then prints it.

```
// code goes here
```

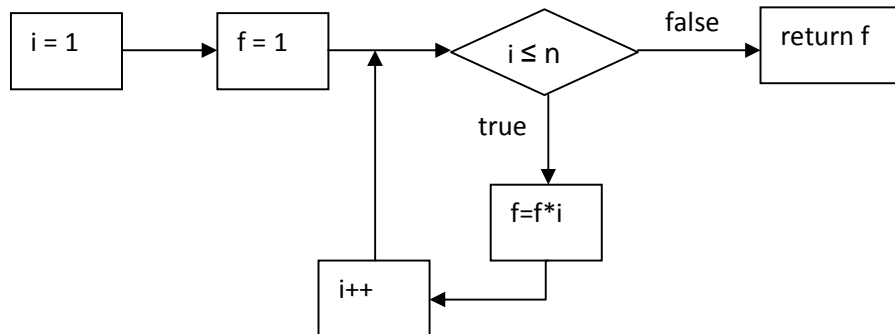
```
double max = 0;
for(int i=0; i<values.size(); i++)
{
    if(values.get(i) > max)
        max = values.get(i);
}
System.out.println(max);
```

Write a **while** loop that examines all the values in an `integer array` called `values`, finds the number of the values that are evenly divisible by 3, stores the result in a variable called `count` & then prints it.

```
// code goes here
```

```
int i=0;
int count=0;
while(i < array.length)
{
    if(array[i] %3 == 0)
        count++;
}
System.out.println(count);
```

Examine the following flow chart & convert into a static method called factorial. 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;
}
```

What does scope mean?

scope refers the region of a program that can refer to/use a variable – typically we talk about local scope (within a block of code) & class scope (usable anywhere inside a class)

What is a precondition? Give an example.

a precondition spells out the requirements of a method in terms of legal variable values (square root method requires a positive input parameter) or object state (withdraw method on a BankAccount would require a sufficient balance)

What does parameter shadowing mean? Give an example.

Parameter shadowing occurs when a method creates a local variable with the same name as an instance field (example on next page)

```
public class Example
{
    private int data;
    public Example(int x)
    {
        int data = x;
    }
}
```

private int data;

public Example(int x)  
{  
 int data = x;  
}

int data = x;

local data variable shadows the instance field of the same name

