QUIZ 2
CMPS 12a - Spring 02
Thomas Raffill

Name: ____________________________
Student ID: ______________________

This exam is closed book, closed notes, no electronic devices. Show all work.
Partial credit given for partial solutions. Presentation counts! Be legible and
coherent for full credit.

Question 1: ____________ (out of 20)
Question 2: ____________ (out of 10)
Question 3: ____________ (out of 10)
Question 4: ____________ (out of 10)
Question 5: ____________ (out of 10)
Question 6: ____________ (out of 10)
Question 7: ____________ (out of 10)
Question 8: ____________ (out of 10)
Question 9: ____________ (out of 10)
Question 10: ____________ (out of 10)

Total: ____________ (out of 100)
1. (10 points)
Assume you already have a function

```java
boolean divisibleBy(int a, int b)
```
which returns true if a is divisible by b, otherwise returns false.

Write a code fragment to check whether a number is divisible by 3, 7 or 13 by calling this function.

The code fragment should print out three lines:
A line saying the number is or is not divisible by 3.
A line saying the number is or is not divisible by 7.
A line saying the number is or is not divisible by 13.

**You don't have to write the function**, just show how to call it.

**Solution:**

```java
if (divisibleBy(x, 3))
    { System.out.println(x + " is divisible by 3."); }
else
    { System.out.println(x + " is not divisible by 3."); }
if (divisibleBy(x, 7))
    { System.out.println(x + " is divisible by 7."); }
else
    { System.out.println(x + " is not divisible by 7."); }
if (divisibleBy(x, 13))
    { System.out.println(x + " is divisible by 13."); }
else
    { System.out.println(x + " is not divisible by 13."); }
```
2. (10 points)
What does this program do? **Hint:** The function names are meaningful.

```java
public class PrintRows {
    public static void main(String[] argv)
    {
        printRow(1);
        printRow(6);
    }

    public static void printRow(int n)
    {
        for (int i = 1; i <= n; i++)
        {
            System.out.print("X");
        }
        System.out.println();
    }
}
```

**Solution:** It prints a row of 1 X followed by a row of 6 X's:

```
X
XXXXXX
```
3. (10 points)
Identify the compiler error in the following code fragment. Explain the error and how to fix it.

```java
public boolean isOdd(int n)
{
if (n % 2 == 0)
{
    boolean k = false;
}
else
{
    boolean k = true;
}
return k;
}
```

**Solution:** The return statement accesses the variable k outside of its scope. There are various ways to fix it. The simplest would be to declare k in the beginning of the function before the conditional branch:

```java
public boolean isOdd(int n)
{
    boolean k;
    if (n % 2 == 0)
    {
        k = false;
    }
    else
    {
        k = true;
    }
    return k;
}
```
Name: 

4. (10 points)
What does the following program print out?

class TrickyScope
{
    public static void main(String[] argv)
    {
        int i = 20, n = 10;
        printNumber(n);
    }

    public static void printNumber(int i)
    {
        System.out.println(i);
    }
}

Solution: 10. The trick is that the variable i in the subroutine refers to the value 10 that was passed to it, and not the i in the main function.
5. (10 points)
Which of the following two boolean expressions will correctly avoid a zero division error when $x = 0$?

(a) $\frac{3}{x} < 5 \quad \text{||} \quad x == 0$

and

(b) $x == 0 \quad \text{||} \quad 3/x < 5$

**Solution:** (b). The expressions are evaluated left-to-right in a lazy fashion. The expression (b) will avoid a zero division error, because if $x = 0$, then the left side of the disjunction is satisfied and the entire expression will evaluate to true without evaluating the right side of the disjunction.
6. (10 points)

What does the following program print out?

```java
class TrickyParameter Passing
{
    public static void main(String[] argv)
    {
        int i = 2;
        int j = f(i) + g(i);
        System.out.println("The answer is "+j);
    }

    public static int f(int x)
    {
        return 3*x;
    }

    public static int g(int x)
    {
        return x*f(x);
    }
}
```

**Solution:** It will print out: The answer is 18

This code computes and assigns to `j` the expression `f(2) + g(2) = 3 \cdot 2 + 2 \cdot (f(2)) = 6 + 2 \cdot 6 = 12 + 6 = 18`. 
7. (10 points)
What does the following code fragment print out?

```java
class TrickyFunction {
    public static void main(String[] argv) {
        int a = 5;
        timesTwo(a);
        System.out.println("The value of a is " + a);
    }

    public static void timesTwo(int a) {
        a = 2*a;
    }
}
```

**Solution:** It prints out: The value of a is 5
The trick is that the variable a in the timesTwo function is only a local copy and does not affect the value of a in the main function.
8. (10 points)
Write a function called isCube which has two int parameters a and b and returns a boolean which is true only if one of them is the cube of the other (in other words, either $a = b^3$ or $b = a^3$) and otherwise is false.

**Solution:** Here is one possible solution for this.

```java
public static boolean isCube(int a, int b) {
    return (a == b*b*b || b == a*a*a);
}
```
9. (10 points)
What does the following program print out?

class TrickyFunctionCall
{
    public static void main(String[] argv)
    {
        int a = 5;
        square(a);
        System.out.println("The value of a is " + a);
    }

    public static int square(int x)
    {
        return x*x;
    }
}

Solution: It prints out: The value of a is 5
The trick is that the expression square(a) does evaluate to 25, but it is not assigned to a and the value of a is unchanged by the function call.
10. (10 points)
What is the value of quizBoolean after the following code fragment executes?

```java
boolean quizBoolean;
int a = 3, b = 100, c = 30, d = 3;
quizBoolean = !(a > b || c % d == 0);
```

**Solution:** The value of quizBoolean is false.

The entire expression is a negation of a disjunction. The expression is true if the disjunction is false. The first part of the disjunction, \(a > b\), is false because it is not true that \(3 > 100\), but the second part of the disjunction, \(c \text{ is true because the remainder of 30 divided by 3 is 0. A disjunction is true if one of its parts is true, so the disjunction is true. Thus, its negation is false.} \)