New York University
Introduction to Computer Science
Exam Sample Problems
© 2013 Andrew I. Case

Instructions:

KEEP TEST BOOKLET CLOSED UNTIL YOU ARE INSTRUCTED TO BEGIN.

This exam is double sided (front and back)!

No calculators, notes, textbooks, or any other aids are allowed except writing utensils (pens, pencils, crayons, erasers, etc.) or other aids provided to you by the instructor. If you need extra scratch paper, please pick it up from the front of the class.

You should also be provided with an appendix that provides helpful documentation.

All answers must be submitted on (or attached to) this exam sheet. All answers must be clearly legible.

Total Points Available:

Total Missed:

Final Score:
Example True or False (10 points):
Instructions: Circle either True or False based on the validity of the statement.

1. Java has a virtual machine comprised of a compiler, linker, and run-time environment.

   True   False

... There are usually 5-10 questions here ...
Example Multiple Choice (10 points):
Instructions: Circle the letter of the best answer.
1. Which of the following keywords will immediately exit a running function?
   A. break
   B. continue
   C. return
   D. goto
   E. None of the above

... There are usually 5-10 questions here ...
Example Short Answer (10 points):

Instructions: In your own words/figures, answer the following as best as possible in one or two sentences.

1. Explain the difference between a float and int data type.

… There are usually about 5 questions here …
Example Entomology - The Study of Bugs (15 points):

Instructions: Find 3 bugs in the following program (there are more than three):

1) Specify the line number of the bug.
2) Check the box next to the type of error; either "Logic" for logic errors or "Other" for (syntax, type, etc.).
3) Write a short explanation or fix for the error.

Note: There can be more than one bug per line.

<table>
<thead>
<tr>
<th>Bug</th>
<th>Line #</th>
<th>Type of Error</th>
<th>Explanation/Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[ ]</td>
<td>Logic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ]</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>[ ]</td>
<td>Logic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ]</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>[ ]</td>
<td>Logic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ]</td>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>
Example What is the Output (15 points):

Instructions: What is the output if the following code is run?

```java
// Global/class variable answer
static int answer = 21;

private static int printAddition(int x, int y) {
    // Local 'answer' variable
    int answer = x + y;
    System.out.println("printAddition():answer = " + answer);
    return answer;
}

private static int subtraction(int x, int y) {
    return x - y;
}

public static void main(String[] args) {
    System.out.println("main(): answer = " + answer);
    printAddition(5,6);
    System.out.println("main(): answer = " + answer);
    System.out.println("main():subtraction(5, 2) = " + subtraction(5, 2));
    System.out.println("main(): answer = " + answer);
}
```

Answer:
Example Document the Code (10 points):

Instructions: The following code was written, but the documentation was left out. Analyze the code to determine what is going on. **Fill in the missing comments (both the /* */ and // parts) to describe what the code is doing.**

```java
/**
 * 
 * 
 * 
 * 
 */
public static boolean isEven(int num) {
    return (num%2 == 0);
}
/**
 * 
 * 
 * 
 * 
 */
public static int factorial(int num) {
    int factorial=1;
    for (int i = 1; i <= num; i++) {
        factorial*=i;
    }
    return factorial;
}
/**
 * 
 * 
 * 
 * 
 */
public static int sumFactorial(int num) {
    int sum = 0;
    for (int i=0; i<=num; i++) {
        sum += factorial(i);
    }
    return sum;
}
```
Example Fill in the Code (10 points):

Instructions: Fill in the missing code. Some program functionality/comments have been omitted for brevity.

```java
import java.util.Scanner;

/**
 * A game that sees how many tries it takes someone to guess a number
 */
public class GuessNumber {
    /**
     * Main program driver
     */
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Default to not being solved
        boolean solved = false;
        // Initialize the number of guesses made
        int numGuesses = 0;
        // Generate a new secret number
        int secretNumber = (int) (Math.random() * Integer.MAX_VALUE);
        // Get guesses from the user until they solved the problem
        while (!solved) {
            // Get next guess
            System.out.print("What is your guess (0-\\n + Integer.MAX_VALUE + ")? ");
            int guess = scanner.nextInt();
            // increment counter
            numGuesses = numGuesses + 1;
            // Did they guess the right number?
            if (guess < secretNumber) {
                System.out.println("Your guess is too low. ");
            } else if (guess > secretNumber) {
                System.out.println("Your guess is too high. ");
            } else {
                solved = true;
                System.out.println("It took you 
 + numGuesses
 + " to guess the secret number; ");
            }
        }
        scanner.close();
    }
}
```
Example Write the Code (20 points):

Instructions: In java code, write a function (feel free to write helper functions) “printSequence” that takes the parameter num and then prints numbers out in according to the sample output:

A call to printSequence(4) should print:

1
21
321
4321
4321
321
21
1

Answer: