COE 211/COE 212 – Computer/Engineering Programming

Welcome to Exam II
Thursday December 20, 2012

Instructor: Dr. George Sakr
Dr. Wissam F. Fawaz
Dr. Maurice Khabbaz

Name: _____

Student ID: _____

Instructions:

1. This exam is **Closed Book**. Please do not forget to write your name and ID on the first page.
2. You have exactly **115 minutes** to complete the 7 required problems.
3. Read each problem carefully. If something appears ambiguous, please write your assumptions.
4. Do not get bogged-down on any one problem, you will have to work fast to complete this exam.
5. Put your answers in the space provided only. No other spaces will be graded or even looked at.

Good Luck!!

Problem 1: comprehension oriented (**20 minutes)** [16 points]
For each question, choose the **single** correct answer.

1) Consider the program segment given below. Its output is :

```
String theWord = "emordnila";
for(int i=0; i < theWord.length()/2; i++)
     System.out.print(theWord.charAt(i));
```

   a. alindrome
   b. emordnila
   c. emord
   d. none of the above

2) Consider the method shown below. How would you best describe its return value:

```
public int method1(int x) {
     int z = 1;
     for(int i = 1; i < x; i++)
          z *= i;
     return z;
}
```

   a. x^x (x raised to x^{th} power).
   b. x^z (x raised to z^{th} power) .
   c. x! (factorial of x).
   d. none of the above

3) Which of the following cannot be used as a parameter for the switch statement?
   a. `int j = 1;`
   b. `boolean k = true;`
   c. `byte b = (byte) 2;`
   d. `char c = 'c';`

4) Consider the following code that will assign a letter grade of 'A', 'B', 'C', 'D', or 'F' depending on a student's test score.

```
if(score >= 90) grade = 'A';
if(score >= 80) grade = 'B';
if(score >= 70) grade = 'C';
if(score >= 60) grade = 'D';
else grade = 'F';
```

   a. This code will work correctly in all cases
   b. This code will work correctly only if score < 70
   c. This code will work correctly only if score < 60
   d. none of the above

5) What is wrong, logically, with the following code?

```
if(x > 10) System.out.println("Large");
else if(x> 6 && x <= 10) System.out.println("Medium");
else System.out.println("Small");
```

   a. The logical error is that no matter what value x is, "Large" is always printed out.
   b. The logical error is that no matter what value x is, "Small" is always printed out.
   c. There is no logical error, but there is no need to have (x<=10) in the second conditional.
   d. There is nothing wrong with the logic at all.

6) The “break” statement does which of the following?
   - **a. Transfers control out of the current control structure such as loop or switch**
   - b. Ends the program
   - c. Denotes the ending of a switch statement
   - d. Indicates the end of line when using System.out.print

7) If x is an `int` where x = 0 initially, what will x be after the following loop terminates?

```
while(x < 100)
      x *= 2;
```

   - **a. This is an infinite loop**
   - b. 64
   - c. 128
   - d. none of the above

8) How many times will the body of the following loop be executed?

```
int x = 10;
while(x > 0) {
      System.out.println(x);
      x--;
}
```

   - a. 9 times
   - **b. 10 times**
   - c. 11 times
   - d. none of the above

9) Assume that x and y are `int` variables with x = 5, y = 3, and a and b are `char` variables with a = ‘a’ and d = ‘A’. Examine the following conditions :

   Condition1: `(x < y && x > 0)`

   Condition 2: `(a != d && ‘9’ > a || d < ‘a’)`

   Condition 3: `(x > y || a == ‘A’ || d != ‘A’)`

   - a. All 3 conditions are false
   - b. Only condition 3 is true
   - **c. Condition 2 and Condition 3 are true only**
   - d. none of the above

10) Which of the following is part of the **java standard class library**?
   - a. Iterator
   - b. Comparable
   - **c. All of the above**
   - d. None of the above

11) The statement `if(x < 0) y = x; else y = 0;` can be rewritten using a conditional operator as:
   - a. (x < 0) ? y = x; y = 0
   - b. x = (x < 0) ? y : 0;
   - c. y = (x < 0) ? 0 : x;
   - **d. none of the above**

12) If x is currently equal to 5, what will be the value of x after following the switch statement executes?

```
switch(x) {
     case 4:
          x += 2;
     case 5:
          x+=3;
     case 7:
          x+=2;
     default:
          x--;
}
```

a. 8
b. 10
c. 11
d. 9

13) Given that `s` is a `String`, what does the following loop do?

```
for(int j = s.length(); j > 0; j--)
     System.out.print(s.charAt(j-1));
```

a. It prints s out forwards
b. It prints s out backwards after skipping the last character of s
c. It prints s out backwards but does not print the first character of s
d. It prints s out backwards

14) The following nested loop structure will execute the innermost statement (x++) how many times?

```
for(int j=0; j < 100; j++)
     for(int k = 100; k > 0; k--)
          x++;
```

a. 10000 times
b. (99 * 99) times
c. 200 times
d. 100000 times

15) Static methods cannot:

a. Reference static instance data
b. Reference non-static instance data
c. Invoke other static methods
d. None of the above

16) In order to implement `Comparable` in a class, what method(s) must be defined in that class?

a. Equals
b. Compares
c. All of the above
d. None of the above

Problem 2: Short true or false questions (**5 minutes)** [10 points]

1. The `break` statement is needed in the last `case` preceding the `default` case in a `switch` selection statement.

Answer: **True** False

2. In Java, it is possible to create an infinite loop out of `while` and `do...while` loops, but not out of `for` loops.

Answer: True **False**

3. The statement `while(a <= b) {a++; b--;}` will do the same thing as the statement `while(!(b < a)) {b--; a++;}`.

Answer: **True** False

4. When an "if" statement has an "else" clause associated with it, the body of the "if" must be delineated with a pair of curly braces.

Answer: True **False**

5. The contents of two strings can be compared using either equality operator (==) or the `equals()` method of the String class.

Answer: True **False**

6. The output of the segment of code shown below is: `sum: 0`.

```
int sum;
for(sum = 0; sum >= 0; sum--)
   sum++;
System.out.println("sum: " + sum);
```

Answer: True **False**

7. An `if-else` statement could be rewritten using two `if` statements without an `else`.

Answer: **True** False

8. The following if statement: `if(name.length() = 3)` can be used to check if the length of the `String` variable called `name` is 3 characters long.

Answer: True **False**

9. Code in Java that uses the && logical operator could be rewritten without &&, using an additional if statement instead.

Answer: **True** False

10. A code in Java that uses the || logical operator can be rewritten without ||, using an additional if statement instead.

Answer: **True** False

Problem 3: Evaluating java expressions (**10 minutes)** [7 points]
For each of the following code fragments, what is the value of x after the statements are executed?

(1)
```
int x=1, i=0;
do {
     x *= i;
     i++;
} while(i < 5);
x += i;
```
Answer: x =**5**

(2)
```
int x = 0;
 for(int i=1; i <= 3; i++)
     for(int j=0; j <= (i/2); j++)
          x += (i+j);
```
Answer: x =**10**

(3)
```
boolean y=true, z=true, w=false; String x="Hi";
x += !(y && z) || (w==y);
```
Answer: x =**Hifalse**

(4)
```
int z=0 w= 1234, x=0;
do {
     z = w%10;
     if(z>x)
     x=z;
     w/=10;
} while(w > 0);
```
Answer: x =**4**

(5)
```
int sum=0, x=0, i=6, j=9;
while(j >= i) {
     sum = sum + j;
     j--;}
x=sum/(j-i);
```
Answer: x =**-30**

(6)
```
int x;
for(int i=1; i<=4; i++)
{int sum = 0; sum = sum + i; x = sum;}
```
Answer: x =**4**

(7)
```
boolean x=true;
for(int i=-2; i<=0; i++)
     if(i < 0)
       x = x && ((2+i) <= 0);
```
Answer: x =**false**

Problem 4: Code Analysis (**12 minutes**) [12 points]

1) Consider the method given below. What would be the output if this were called using the statement: `method1();`?

```
void method1(){
      int x = 13572 ;
      while(x > 10)
            x = x/10 ;
      System.out.print(x) ;
}
```

a. 2
b. 27531
c. 1357
d. none of the above

2) Consider the code segment given below. Its output is:

```
int x = 1357;
int z = 0;
for(int i=x; i>0; i=i/10)
      z = z + (i%10);
System.out.print(z);
```

a. 7531
b. 156
c. 16
d. none of the above

3) Consider the method given below. What would be the output if this were called using the statement: `System.out.println(method2(65432);`?

```
void method2(int number){
      int x = number;
      int y = 0;
      while(x > 0) {
            y = y*10 + x % 10;
            x = x/10 ;
      }
      return y ;
}
```

a. 12345
b. 5
c. 23456
d. 1

Problem 5: Method definition (**15 minutes)** [9 points]

You are given below the headers of 3 methods called `reverseString`, `removeNonAlphabets`, and `isPlaindromePhrase`. Your job is to complete the definition of each one of these methods as per the provided guidelines.

1. **`reverseString()`** is a method that accepts a String and returns a reversed version of the input String.

```
public String reverseString(String s) {
     String str = new String(“”);
     for (int i = s.length()-1; i >= 0; i--)
          str += s.charAt(i);
     return str;
}
```

2. **`removeNonAlphabets()`** is a method that accepts a `String` parameter and returns the same string with all non-alphabetic characters removed.

```
public String removeNonAlphabets(String word) {
     String str = new String(“”);
     char letter;
     for (int j = 0; j <= word.length()-1; j++)
     {
          letter = word.charAt(j);
          if ((letter >= ‘A’ && letter <= ‘Z’) ||
              letter >= ‘a’ && letter <= ‘z’))
               str += letter;
     }
     return str;
}
```

3. **`palindromePhrase()`** is a method that accepts a String parameter and returns true if the string is a palindrome and false otherwise. However, all non-alphabetical characters are ignored. For example, “Lonely Tylenol” and “Madam, I’m Adam” are both considered to be palindromes. This method must make use of the 2 above-presented methods.

```
public boolean palindromePhrase(String phrase) {
     phrase = removeNonAlphabets(phrase);
     String reversedPhrase = reverseString(phrase);
     return(phrase.equals(reversedPhrase));
}
```

Problem 6: Debugging Problem (**10 minutes)** [6 points]

Consider the following program, which is supposed to print out the powers of 3 from 3^0 up to and **including** 3^N, where N is a non-negative integer that is entered by the user.

```
1    import java.util.Scanner;
2    public class PowersOfThreeBuggy {
3         public static void main(String[] args){
4              Scanner scan = new Scanner(System.in);
5              int N = scan.nextLine();
6              int i = 1;
7              int base = 3;
8              while (i < N){
9              System.out.println("3^" + i +
10                        " = "+ Math.pow(base, i));
11             i++;}
12   }}
```

This program has three bugs.

A. Which bug prevents the program from *compiling* successfully? Identify the line number where the bug appears and give a correct version of this line of code.

Line number **_5____**

Correct version: **__int N = scan.nextInt();_____**

B. Identify the line numbers where the two *logical* errors appear and give a correct version of each line of code.

Line number **__6___**

Correct version: **__int i = 0;_____**

Line number **__8___**

Correct version: **__while(i <= N)_____**

Problem 7: Coding (**40 minutes)** [40 points]

1. Write a program named `EvenDigitAv` that reads an integer from the user and prints the average of all its **even digits.** For example if the user enters 158621, your program will display the average of its even digits namely, 8, 6 and 2.

Sample output

```
Enter a number: 269914
The average of all the even digit in 269914 is: 4.0
```

```
import java.util.Scanner;

public class EvenDigitAv {
     public static void main (String[] args) {
          Scanner scan = new Scanner (System.in);
          System.out.print(“Enter an integer: ”);
          int num = scan.nextInt();
          int sum = 0; int count = 0;
          while (num > 0) {
               int d = num % 10;
               if (d % 2 == 0) {
                    count++;
                    sum += d;
               }
               num /= 10;
          }
          System.out.println(“Average of even digits is: “+
                                        (double) sum/count);
     }
}
```

2. Write a program named `CountCaps` that reads a String from the user and prints the number of uppercase letters and the number of lowercase letters found in the string.

Sample output

```
Enter a sentence: This is a FUN EXAM
Uppercase count:  8
Lowercase Count: 6
```

```
import java.util.Scanner;

public class CountCaps {
     public static void main (String[] args) {
          Scanner scan = new Scanner (System.in);
          System.out.print (“Enter a string: “);
          String str = scan.nextLine();
          int countUp = 0, countLow = 0;
          char letter;
          for (int j = 0; j <= str.length() – 1; j++)
          {
               letter = str.charAt(j);
               if (letter >= ‘A’ && letter <= ‘Z’)
                    countUp++;
               if (letter >= ‘a’ && letter <= ‘z’)
                    countLow++;
          }
          System.out.println(“Uppercase count: “ +
                              countUp);
          System.out.println(“Lowercase count: “ +
                              countLow);
     }
}
```

3. Write a program called `SumOfPows` that reads from the user an integer base `b` and a maximum integer exponent `n` and then prints the sum of all the powers of `b` from 0 up to `n` inclusive (i.e. $b^0 + b^1 + b^2 \ldots + b^n$).

Sample output

```
Enter base: 2
Enter Max Exponent: 3
The sum of the powers of 2 is: 15
```

```
import java.util.Scanner;

public class SumOfPows {
     public static void main (String[] args) {
          Scanner scan = new Scanner (System.in);
          System.out.print("Enter an integer base: ");
          int b = scan.nextInt();
          System.out.print("Enter an integer exponent: ");
          int n = scan.nextInt();
          int sum = 0;
          for (int j = 0; j <= n; j++)
               sum += (int) Math.pow(b,j);
          System.out.println("The sum of the powers of"+b+
                                   "is: " + sum);
     }
}
```

4. Write a program to read a list of exam scores given as integers stored in a file called “grades.txt”. These grades are in the range of 0 to 100. The program has to write the following to a file called “gradeStat.txt”:
    1. The total number of grades that was read.
    2. The average of these grades.
    3. The number of grades in each letter-grade category as follows:
        - A: 90 to 100.
        - B: 80 to 89.
        - C: 70 to 79.
        - D: 60 to 69.
        - F: 0 to 59.

```
import java.util.Scanner;
import java.io.*;

public class ExamScores {
     public static void main (String[] args) {
          File f1 = new File(“grades.txt”);
          Scanner fileScan = new Scanner(f1);
          File f2 = new File(“gradeStat.txt”);
          FileWriter fw = new FileWriter(f2);
          BufferedWriter bw = new BufferedWriter(fw);
          PrintWriter pw = new PrintWriter(bw);
          int countA = 0, countB = 0, countC = 0, sum = 0,
              countD = 0, countF = 0, countTotal = 0,
          while (fileScan.hasNext()) {
               String gradeLine = fileScan.nextLine();
               Scanner gradeScan = new Scanner(gradeLine);
               while (gradeScan.hasNext()) {
                    countTotal++;
                    int grade = Integer.parseInt(
                                   gradeScan.next());
                    sum += grade;
                    switch(grade / 10){
                         case 9: countA++; break;
                         case 8: countB++; break;
                         case 7: countC++; break;
                         case 6: countD++; break;
                         default:
                              if (grade == 100) countA++;
                              else countF++;
                    }
               }
          }
          pw.println(“The total number of grades is: “ +
                         countTotal);
```

```
            pw.println("The average is: " +
                            (double) sum/countTotal);
            pw.println("A: " + countA);
            pw.println("B: " + countB);
            pw.println("C: " + countC);
            pw.println("D: " + countD);
            pw.println("F: " + countF);
            pw.close();
        }
    }
```