COE 211/COE 212 – Computer/Engineering Programming

Welcome to Exam II
Friday January 11, 2012

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

Name: _____Solution Key_______

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 **120 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!!

<u>Problem 1:</u> comprehension oriented (**30 minutes)** [15 points]

For each question, choose the **single** correct answer.

1) If `category` is a variable of type `String` and whose value is `“laptop”`, which of the following conditions is true:

   a. `category = “laptop”`
   b. `category == “laptop”`
   c. `category == new String(“laptop”)`
   d. `category.equals(“Laptop”)`
   e. **None of the above**

2) Let `i` be an integer variable that has been initialized to the value `2`. Then the execution of the statement `int j = (i++) + (++i) + (i++);` will cause the two-tuple `(i,j)` to be:

   a. (5,9)
   b. **(5,10)**
   c. (5,11)
   d. (5,12)
   e. (4,10)

3) Let `b` be a variable of type `byte`. The compilation of which of the below statements will result in an error:

   a. `b++;`
   b. `b--;`
   c. **`b = b + 1;`**
   d. The second and third of the above statements.
   e. All of the above statements.

4) Consider the following piece of code:

```
for (int i = 0; i<2; i++) {
     switch(i) { case 0: System.out.print("v ");break;
                 case 1: System.out.print("w ");
                 case 2: System.out.print("x ");break;
                 case 3: System.out.print("y ");
                 case 4: System.out.print("z ");break;
                default: System.out.print("d ");}}
```

When the above code is executed, the output will be:

   a. v w y
   b. v w x
   c. v w z
   d. v w w
   e. d d d

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

```
if(x > 10); System.out.println(“Large”);
if(x <= 10 && x > 6) 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. 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) Consider the following code:

```
    double x; int a = 10;
    if (a >= 0) x = 3.14;
    if (a <  0) x = 2.71;
    System.out.println(x);
```

Execution of the above code will cause:

a. **A compile-time error to occur.**
b. A run-time error to occur.
c. A logical error to occur.
d. None of the above.

7) What will `x` be after the following loop terminates?

```
for (int x = 1; x > 0; x--)
    x *= 2;
```

a. **This is an infinite loop**
b. 0
c. 1
d. None of the above

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

```
int x = 11;
while(x >= 1) {
    System.out.println(x);
    x--;
}
```

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

9) Assume that, in the code below, `N` and `x` are integers with `N > 1`, `x = 1`.

```
    while (N >= 1) {
      System.out.println(x);
      x = 2 * x;
      N = N / 2;}
```

The execution of this code will :

a. **print out all of the powers-of-two that are less than or equal to `N`.**
b. print out all of the powers-of-two that are greater than 1.
c. print out the powers-of-two including 2^0 and 2^N.
d. All of the above.

10) Assume that val has been defined as an int for the code below:

```
if (val>4) System.out.println("Test A");
else if (val>9) System.out.println("Test B");
else System.out.println("Test C");
```

What is the set of all values that `val` can take so that only "Test C" is printed?

a. `val <= 4`
b. `0 < val && val < 4`
c. `4 < val && val < 9`
d. `val > 9`
e. None of the above

11) Consider the following piece of code:

```
do{y = x + 7; x++;}while(x < 9);
```

Which of the following pieces of code are equivalent to the above code:

a. `y = x + 7; x++; while(x < 9){y = x + 7; x++;}`

b. `while(x < 9){y = x + 7; x++;} y = x + 7; x++;`

c. `while(x <= 9){y = x + 7; x++;}`

d. Choices a and b.

12) The `continue` statement transfers the flow of control:

a. to the initialization statement.

b. to the conditional statement of the `for` loop.

c. to the update statement of the `for` loop.

d. to the statement just after the `for` loop.

e. to the statement just before the `for` loop.

13) Assuming that `i` is initialized to `3` then the following nested loop structure will execute the innermost statement `x++` how many times?

```
while(i > 0){
     j = 4;
     while(j >= 1) { x++; j--; }
     i-=2;}
```

a. 6 times

b. (3 * 4) times

c. 8 times

d. 7 times

14) Which of the following statements is true?

a. A static variable must be declared as final.

b. An instance variable cannot be declared final.

c. A static method cannot access a non-static instance variable.

d. Only a static method can reference a static variable.

15) In order to implement `Iterator` interface in a class, what method(s) must be defined in that class?

a. `next()`

b. `nextLine()`

c. `hasNext()`

d. Choices a and c

e. All of the above

Problem 2: Short true or false questions (**7 minutes)** [10 points]

1. The `Comparable` interface offers two methods, namely the `equals()` and `equalsIgnoreCase()` methods.
Answer: True **False**

2. The `do...while` loop executes its body at most once before it evaluates the conditional expression.
Answer: **True** False

3. The output of the following statements is: 10 Done
```
int y = 10, z = 0;
System.out.println(“” + z + y + “Done”);
```
Answer: True **False**

4. The statement `if(a||b) x++;` is equivalent to the combination of the two statements `if(a) x++; if (b&&!a) x++;`
Answer: **True** False

5. An empty `for` loop such as `for ( ; ; )` is interpreted as a syntax error.
Answer: True **False**

6. The output of the segment of code shown below is: `i = 1`.
```
int i = 0;
do{
     i++;
} while(i-- == 0);
System.out.println(“i = “ + i);
```
Answer: True **False**

7. An object that is made up of a collection of other objects is called an aggregate object.
Answer: **True** False

8. The following `if` statement: `if(name1.comparTo(name2) = 0)` indicates whether or not the `String name1` is equal to the `String name2`.
Answer: True **False**

9. If The evaluation of the condition `(z+1>=3 || x+1<z*10 && y+3>7)` yields `true` given that `x = 10, y = 5, z = 2`.
Answer: **True** False

10. A `switch` statement with three cases is equivalent to an expression involving the ternary conditional operator `_ ? _ : _`
Answer: True **False**

<u>Problem 3:</u> Evaluating java expressions **(13 minutes)** [7 points]
For each of the following code fragments, what is the value of x after the statements are executed?

```
(1)                 String name = “Bob F. Dag”;
  boolean startWord = true;
  String x = “”;
  for(int i = 0; i < name.length(); i++){
       if (startWord) x += name.charAt(i);
       if (name.charAt(i) == ‘ ’) startWord = true;
       else startWord = false;}
```

Answer: **x = B F D**

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

Answer: **x = 13**

```
(3)                 int y = -32.2; String x;
  boolean isPositive = (y > 0);
  if (isPositive = true) x = “positive”;
  else x = “not positive”;
```

Answer: **x = “positive”**

```
(4)                 int x=0;
  for (int i = 0; i < 2; i++)
  for (int j = 0; j < 2; j++)
  for (int k = 0; k < 2; k++)
  if (2*i + j >= 3*k) x++;
```

Answer: **x = 5**

```
(5)                 int x = 10;
  x = x++; x = ++x;
  x = (x++) + (++x);
```

Answer: **x = 24**

```
(6) int x = 0, y = 1;
  for(int i = 0; i <= 5; i++)
  {x += y; y = x – y;}
```

Answer: **x = 8**

```
(7)                 String sentence = “Hello There”;
  String str1 = sentence.replace(‘e’, ‘l’);
  String str2 = sentence.replace(‘l’, ‘e’);
  boolean x = sentence.equalsIgnoreCase(str2);
```

Answer: **x = false**

<u>Problem 4:</u> Code Analysis (**15 minutes**) [12 points]

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

```
public void method1(){
     int i = 0, j = 5;
     for( ; (i < 3) && (j++ < 10) ; i++ )
          System.out.print(" " + i + " " + j );
     System.out.print(" " + i + " " + j );
}
```

a. 0 6 1 7 2 8 3 9
b. 0 6 1 7 2 8 3 8
c. 0 5 1 5 2 5 3 5
d. none of the above

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

```
int Output = 10;
boolean b1 = false;
if(b1==true && (Output += 10)==20)
   System.out.println("We are equal " + Output);
else
   System.out.println("Not equal! " + Output);
```

a. We are equal 20
b. We are equal 10
c. Not equal! 10
d. Not equal! 20
e. 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(654321);`?

```
public String method2(String num){
     String modNum = “”;
     for(int i = num.length()-1; i>0; i--)
          modNum = modNum.concat(“” + num.charAt(i));
     return modNum;
}
```

a. 123456
b. 2345
c. 23456
d. 12345

<u>Problem 5:</u> Method definition (**15 minutes)** [12 points]

Complete the definition of each one of these methods as per the provided guidelines.

1. **check()** is a method that accepts a String called word and returns true if the last three characters word are “yay”. Otherwise, the method returns false.

```
public static boolean check(String word) {

     String subword = word.substring(word.length()-3,
                                     word.length());
     boolean result = subword.equalsIgnoreCase("yay");
     return result;
}
```

2. **extractOriginal()** is a method that accepts a String called word as a parameter and prints out a modified version of word excluding its last three characters.

```
public static void extractOriginal(String word) {
     System.out.println(word.substring(0,word.length()-3));
}
```

3. **convertToOriginal()** is a method that accepts a String called word as a parameter and prints out a modified version of word in which the last two characters are removed and the third last character is placed at the beginning. For example, if word = “moody”, then the modified word becomes “omo”.

```
public static void convertToOriginal(String word) {
     System.out.println("" + word.charAt(word.length()-3) +
                    word.substring(0,word.length()-3));
}
```

4. **translate()** is a method that accepts a Pig Latin String called sentence as a parameter and prints out its corresponding translated original English version. Note that a sentence is composed of multiple words and this method **<u>must make use</u>** of the three above-defined methods.

```
public static void translate(String sentence) {
     Scanner scan = new Scanner(sentence);
     while(scan.hasNext()){
          String word = scan.next();
          if(check(word))
               extractOriginal(word);
          else
               convertToOriginal(word);
     }
}
```

<u>Problem 6:</u> Coding (**40 minutes)** [44 points]

1. Write a program called `Eureka` that reads an integer `n` from the user and prints out to the screen the word Eureka if both the sum of the **<u>odd digits</u>** of `n` is even and the sum of the **<u>even digits</u>** is less than or equal to 20**.**

<u>Sample output</u>

```
Enter a number: 1586213
The sum of the odd digits is: 10.
The sum of the even digits is: 16.
Eureka
```

```
import java.util.Scanner;

public class Eureka{
     public static void main (String[] args){
          Scanner scan = new Scanner(System.in);
          System.out.print(“Enter an intger: “);
          int n = scan.nextInt();
          int sumOdd = 0, sumEven = 0;
          while(n > 0){
               int x = n%10;
               if (x%2 == 0)
                    sumEven += x;
               else
                    sumOdd += x;
               n /= 10;
          }
          if (sumOdd % 2 == 0 && sumEven <= 20)
               System.out.println(“Eureka”);
     }
}
```

2. Write a program named `SumDigits` that reads a line of text from the user. The text line may include both alphabetical letters from ‘a’ to ‘z’ (either capital or lower case) and numerical digits from 0 to 9. The program will then print out to the screen the number of digits found in the text line as well as their sum.

Sample output

```
Enter a text line: ayee87rti85po9kjs5j3k1ljk2q6
The number of digits is:  10
The sum of digits is: 54
```

```
import java.util.Scanner;

public class SumDigits {
     public static void main (String[] args){
          Scanner scan = new Scanner(System.in);
          System.out.print(“Enter a line of text: “);
          String line = scan.nextLine();
          int sum = 0, count = 0;
          for (int i = 0; i < line.length(); i++){
               char c = line.charAt(i);
               if (c >= ‘0’ && c <= ‘9’){
                    int val = Integer.parseInt(“” + c);
                    sum += val; count++;
               }
          }
          System.out.println(“The number of digits is: “ +
                                                  count);
          System.out.println(“The sum of digits is: “ + sum);
     }
}
```

3. Write a program called `RemoveChar` that reads from the user a sentence `s` and a character `c` and then prints the modified sentence where all instances of the character `c` are removed.

Sample output

```
Enter sentence: hello how are you. I am fine thank you
Enter a character to remove: o
The modified sentence is: hell hw are yu. I am fine thank yu.
```

```
import java.util.Scanner;

public class RemoveChar{
    public static void main(String[] args){
           Scanner scan = new Scanner(System.in);
           System.out.println("Enter a sentence:");
           String str = scan.nextLine();
           System.out.println("Enter the character you want to” +
                                       remove from the sentence: ");
           String letter = scan.nextLine();
           char c = letter.charAt(0);
           int index = str.indexOf(c);
           while (index != -1){
                String part1 = str.substring(0,index);
                String part2 = str.substring(index+1,str.length());
                str = part1 + part2;
                index = str.indexOf(c);
           }
           System.out.println("The modifed sentence is:");
           System.out.println(str);
    }
}
```

4. Write a JAVA program named `Duplicate` that creates a copy of a file called `original.txt`. The copy will be called `copy.txt`. Sometimes duplication errors may occur. Hence, after creating the copy, your program has to verify whether or not the original file and its copy have the same content. This verification process is done line by line. Your program will print the message SUCCESS if the two files have similar contents and FAILURE otherwise.

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

public class Duplicate{
     public static void main(String[] args){
          File orgFile = new file(“original.txt”);
          File copFile = new file(“copy.txt”);
          FileWriter fw = new FileWriter(copFile);
          BufferedWriter bw = new BufferedWriter(fw);
          PrintWriter pw = new PrintWriter(bw);
          Scanner fileScan = new Scanner(orgFile);
          while(fileScan.hasNext()){
               String line = fileScan.nextLine();
               pw.println(line);
          }
          pw.close();
          File verFile = new file(“copy.txt”);
          Scanner verScan = new Scanner(verFile);
          Scanner orgScan = new Scanner(orgFile);
          while(orgScan.hasNext()){
               String orgLine = orgScan.nextLine();
               String verLine = verScan.nextLine();
               if(orgLine.compareTo(verLine) != 0){
                    System.out.println(“FAILURE”);
                    System.exit(1);
               }
          }
          System.out.println(“SUCCESS”);
     }
}
```