COE 212 – Engineering Programming

Welcome to Exam II
Tuesday May 13, 2014

Instructors: Dr. Georges Sakr
Dr. Joe Tekli
Dr. Wissam F. Fawaz

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 **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!!

Problem 1: Multiple choice questions (**15 minutes)** [10 points]

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

1) What is the output of the code fragment given below?

   ```
   int count=20; boolean done=false; double sum=123.5;
   if((done && count>0) || !done || sum<0)
   if(sum > 0)
   System.out.println(“Sum is positive”);
   else
   System.out.println(“Sum is not positive”);
   else
   done = !done;
   System.out.println(“Sum is: ” + ((done) ? sum : 10*sum));
   ```

   a. Sum is positive
   b. Sum is not positive
   c. Sum is positive
      Sum is: 123.5
   d. **None of the above**

2) The statement: `System.out.println(str.compareTo(“Hello”));` produced a positive numeric value. Which of the following is/are possible values of the `String` object `str`?

   a. **`“hello”`**
   b. `‘1’`
   c. `“Hell”`
   d. Both (b) and (c)

3) Which of the following statements checks to see whether the last character in the `String` variable called `str` is equal to the letter h?

   a. `str.charAt(str.length()-1).equalsIgnoreCase(“H”)`
   b. `str.charAt(str.length()-1) == “h”`
   c. Both of the above
   d. **None of the above**

4) Which of the following repetition statements would execute the body of the loop **10 times**?

   a.
   ```
   int i = 1;
   while(true) {
         i++;
         if(i== 11)
               break;
   }
   ```
   b.
   ```
   int i=1;
   while(i<=11) {
         if(i==2)
               continue;
         i++;}
   ```
   c. Both of the above
   d. None of the above

5) Which of the following statements increments the variable called `sum` by `2` when the variable called `val` is **strictly greater** than `MAX` and by `3` **otherwise**?

   a. `sum = sum + ((val<=MAX)?3:2);`
   b. `sum += (MAX<val)?2:3;`
   c. **Both of the above**
   d. None of the above

6) Consider the following `switch` statement. What value would be stored in `sum` after it is executed?

```
char choice=‘A’; int sum=1;
switch(choice) {
case ‘B’:
      sum*=2;
case ‘A’:
      sum*=3;
case ‘C’:
      sum*=4;
default:
      sum*=5;
}
```

   a. 3
   b. 12
   c. **60**
   d. None of the above

7) What output does the following program produce?

```
public class Test {
      public static void main(String[] args) {
            int max = 0;
            max(1, 2, max);
            System.out.print(max);}
      public static int max(int value1, int value2, int max) {
            if (value1 > value2)
                  max = value1;
            else
                  max = 5*value2;
            return max;
      }}
```

   a. 2
   b. 5
   c. **0**
   d. None of the above

8) After execution of the following piece of code, the value of `rubbish` will be:

```
String str = “Exam is fun”; int rubbish=0;
for(int i=0; i<=str.length()/2; i++){rubbish += i%2;}
```

   a. 4
   b. **3**
   c. 2
   d. None of the above

9) Which of the following **is not** part of the Java standard class library?
   a. `integer`
   b. `Char`
   c. **Both of the above**
   d. None of the above

10) `switch` statements are distinguished from typical `if-else` statements in that:
   a. They can only test for byte, short, integer, boolean, character, and `String` literals.
   b. **`switch` cases are not mutually exclusive, that is, more than one case can be executed at a time.**
   c. Both of the above.
   d. None of the above.

<u>Problem 2:</u> True or false questions (**15 minutes)** [12 points]

1. The output of the following nested for loops will be as shown in the box given below.

```
for (int i = 0; i < 4; i++) {
   for (int j = 0; j< (4-i); j++)
      System.out.print("*");
System.out.println();}
```

```
****
***
**
*
```

Answer: **True** False

2. The following code in Java stores in variable `prod` the product of the integer values between 1 (inclusive) and 10 (inclusive).

```
int prod = 1; int x;
for(x=1; x<11; x++)
      prod *= x;
```

Answer: **True** False

3. If a class implements `Comparable`, then an object created from that class can invoke the `compareTo` method.

Answer: **True** False

4. Private instance variables can only be directly accessed through code in the class where they are declared.

Answer: **True** False

5. A non-static method can access static instance variables.

Answer: **True** False

6. The output of the program segment given below is: `8`

```
int x = 1;
for(int i=1; i<3; i++)
for(int j=i; j<3; j++)
x = x*j;
x+=x;
System.out.print(x);
```

Answer: **True** False

7. Consider the following sequence of `if` statements:

```
if(a < b)
  if(c < d)
      answer = 2;
if(e < f)
      answer = 2;
```

   The series of `if` statements above can be rewritten as follows:

```
 if((a < b) || (c < d) && (e < f))
      answer = 2;
```

Answer: True **False**

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

```
int sum=0;
for(sum=-1; sum >= 0; sum--)
      sum++;
 System.out.println(“sum: ” + sum);
```

Answer: True **False**

9. The following can be used to check to see if the value of a variable `x` is divisible by 6: `if(!(x%2!=0) || (x%3==0))`

Answer: True **False**

10. The following code prints the even numbers from 1 (inclusive) to 20 (inclusive), each on a different line.

```
int i = 1;
while(i<=20) {
      i++;
      System.out.print(i);
      if(i%2==0)
            System.out.println();
      i++;
}
```

Answer: **True** False

<u>Problem 3:</u> Code analysis (**15 minutes**) [10 points]

1) Consider the methods given below. What would be returned if it were called using the statement: `method1(“relentless”);`?

```
public String method1(String theWord){
      char c;
      String newWord= “”;
      for(int i=0; i<theWord.length(); i++){
            c = theWord.charAt(i);
            newWord = “”;
            for(int j=0;j<theWord.length();j++){
                  if(j<=i) {
                        newWord=newWord+theWord.charAt(j);
                  } else if(theWord.charAt(j)!=c){
                        newWord=newWord+theWord.charAt(j);
                  }
            }
            theWord=newWord;
      }
      return newWord;}
```

a. sseltneler
b. relntles
c. relnts
d. Compile-time error
e. none of the above

2) What would be the output of the method call: `method2(14679);`

```
public void method2(int number){
      int x = number;
      int count = 0;

      while(x>0) {
            x = x/10;
            count++;
      }

      for(int i=0; i<count/2; i++) {
            number = number/10;
      }

      System.out.print(number%10);
}
```

a. 9
b. 1
c. 6
d. Compile-time error
e. None of the above

Problem 4: Evaluating java expressions **(15 minutes)** [8 points]

For each of the following code fragments, what is the value of x after the statements are executed?

(1)
```
int x=2, i=0;
do {
x += i;
i++;
} while(i < 5);
x -= i;
```

Answer: **x = 7**

(2)
```
int z=0, w= 123, x=0;
do {
z = w%10;
if(z>x)
x=z;
w/=10;
} while(w > 0);
```

Answer: **x = 3**

(3)
```
char choice = ‘D’; int x = 10;
switch(choice) {
     case ‘B’:
      x += 10;
case ‘D’:
      x += 20;
case ‘C’:
      x += 30;
default:
      x += 40;}
```

Answer: **x =100**

(4)
```
boolean x=false;
for(int i=-2; i<=2; i++)
if(i%2 == 0)
x = ((2+i) <= 0) || x;
```

Answer: **x =true**

(5)
```
int x=0, s;
for(s=++x; s<5; ++s)
x+=s;
s++;
```

Answer: **x =11**

(6)
```
int number=1324; int x=number, count=0;
while(x>0) {
      x=x/10; count++;}
for(int i=0; i<count/2; i++)
      number = number/10;
x=number%10;
```

Answer: **x =3**

(7)
```
int N, P; boolean x;
N=5; P=2; x = N++ > P || P++ != 3;
```

Answer: **x =true**

Problem 5: Program Analysis (**10 minutes**) [10 points]

Consider the classes *Stock* and *ManipulatingStock* given below:

```
public class Stock {
   private int numItems;
   public Stock(int numItems){ this.numItems = numItems;}
   public String toString(){ return "" + numItems; }
   public static void Add(int n1, int n2) {
      int S = n1 + n2;
      System.out.println(" Sum of " +n1+ " and " +n2+ " = " + S);
      n1 = S++;
      n2 = S++;
    }
   public static void Add(int n1, Stock n2) {
      int S = n1 + n2.numItems;
      System.out.println("Sum of " +n1+ " and " +n2+ " = " +S);
      n1 = S++;
      n2.numItems = S++;
   } }
```

```
public class ManipulatingStock{
   public static void main(String[] args){
   Stock s1 = new Stock(10);
   Stock s2 = new Stock(30);
   int s3 = 20;
   System.out.println("s1=" +s1+ " s2=" +s2+ " s3=" +s3);
   Stock.Add(s3, s1);
   System.out.println("s1=" +s1+ " s2=" +s2+ " s3=" +s3);
   Stock.Add(s3, s2);
   System.out.println("s1=" +s1+ " s2=" +s2+ " s3=" +s3);
   }}
```

Provide in the box below the output that would be displayed on-screen when the class `ManipulatingGrades` is executed:

```
s1=10  s2=30  s3=20
Sum of 20 and 10 = 30
s1=31  s2=30  s3=20
Sum of 20 and 30 =50
s1=31  s2=51  g3=20
```

<u>Problem 6:</u> Method definition (**10 minutes)** [12 points]

You are given below information about 3 methods called **`isPrime`**, **`isPalindrome`** and **`isPalindromeAndPrime`**. Your job is to complete the definition of each one of these methods as per the provided guidelines.

1. **`isPrime()`**is a method that accepts a single `int` parameter `n`, and returns true if `n` is prime and false otherwise. Note that a number is considered to be prime if it is divisible only by 1 and itself.

```
public boolean isPrime(int n) {
     boolean output = true;
     for(int i=2; i<=n/2; i++) {
          if(n%i==0) {
               output = false;
               break;
          }
     }
     return output;}
```

2. **`isPalindrome()`** is a method that accepts a `String` object `str` and returns true if `str` is a Palindrome and false otherwise. Note that a String is considered to be a Palindrome if it reads the same forward and backward.

```
public boolean isPalindrome(String str) {
     String reverse = “”;
     for(int i=str.length()-1; i>=0; i--)
          reverse+=str.charAt(i);
     return reverse.equalsIgnoreCase(str);
}
```

3. **`SumPrimes()`** is a method that accepts an `int` parameter `n` and returns the sum of all the prime numbers that are less than or equal to `n`. You should use the **`isPrime()`** method defined earlier.

```
public int sumPrimes(int n) {
     int sum=0;
     for(int i=1; i<=n; i++) {
               if(isPrime(i))
                    sum+=i;
     }
     return sum;
}
```

<u>Problem 7:</u> Coding (**40 minutes)** [40 points]

1. Write a program called `ComputingSeries` which reads from the user a strictly positive (> 0) integer `n`, and then computes the following sum: $S = 1^{n} + 2^{n-1} + 3^{n-2} + \ldots + i^{n-i+1} + \ldots + n^{1}$. Note that the program has to make sure that the number `n` provided by the user is <u>positive</u>.

<u>Sample output</u>
Enter a positive number: -2
The number you entered is not positive. Please try again: 4
S: 22

```
import java.util.Scanner;
public class ComputingSeries {
      public static void main(String[] args) {
            int n, S=0;
            Scanner scan = new Scanner(System.in);

            System.out.println(“Enter a positive number:”);
            n = scan.nextInt();

            while(n < 0) {
                  System.out.println(“The number you entered is ”
                  + “not positive. Please try again:”);
                  n = scan.nextInt();
            }

            for(int i=1; i<=n; i++)
                  S+=(int) Math.pow(i, n-i+1);
            System.out.println(“S: ”+S);
      }
}
```

2. An integer number is said to be perfect if its divisors, including `1` (but not the number itself), sum up to the number. For example, `6` is a perfect number, because `6=1+2+3`, with `1`, `2`, and `3` being the factors of `6`. Write a program called `PerfectNumbers` that asks for and reads from the user an **integer value** denoted by `X`. The application should print all the perfect numbers between `1` and `X`.

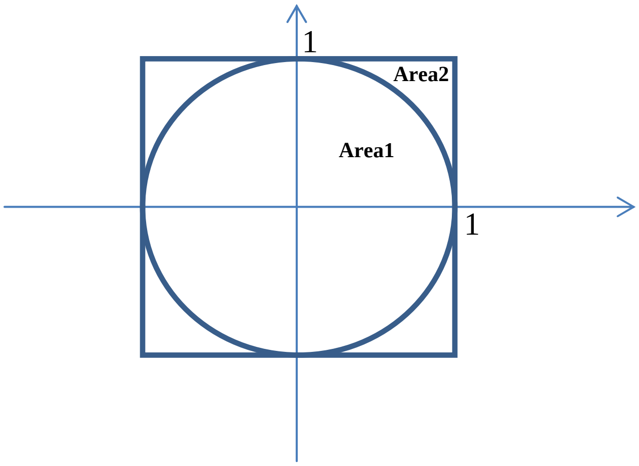
<u>Sample output</u>:

```
Enter value: 28
Perfect numbers between 1 and 28:
6
28
```

```
import java.util.Scanner;
public class PerfectNumbers {
public static void main(String[] args) {
      int X, sum;
      Scanner scan = new Scanner(System.in);

      System.out.println(“Enter value:”);
      X = scan.nextInt();
      System.out.println(“Perfect numbers between 1 and ” + X);
      for(int i=1; i<=X; i++) {
            sum = 0;
            for(int j=1; j<i; j++)
                  if(i%j==0)
                        sum+=i;
            if(sum==i)
                  System.out.println(i);
      }
}
}
```

3. Write a program called `MonteCarlo` that reads an integer `n` from the user. Your program must then loop `n` times and in every iteration through the loop body, your program has to generate 2 random double values `x` and `y` between 0 (inclusive) and 1 (exclusive). Every `(x, y)` pair of values that are generated corresponds to a point in the plane that lies either in **Area1** or **Area2** of the figure given below. Your program should count the number of points that belong to **Area1**, where $x^2+y^2<1$. Your program should then print the ratio between the **number of points found in Area1** and the **total number of generated points** (`n`).



```
import java.util.Scanner;
import java.util.Random;
public class MonteCarlo {
      public static void main(String[] args) {
            int n, nbArea1=0;
            double x, y;
            Scanner scan = new Scanner(System.in);
            Random rnd = new Random();

            System.out.println("Enter n:");
            n = scan.nextInt();

            for(int i=1; i<=n; i++) {
                  x = rnd.nextFloat();
                  y = rnd.nextFloat();

                  if((x*x + y*y) < 1.0)
                        nbArea1++;
            }

            System.out.println("Ratio: " + (double) nbArea1/n);
      }
}
```

4. Write a program called `FileStats` to read the contents of a text file called `“test.txt”`. The program has to print out the following:
    1. The total number of words in the file.
    2. The ratio between the total number of vowels and the total number of characters found in the file.

```
import java.util.Scanner;
import java.io.*;
public class FileStats {
      public static void main(String[] args) throws IOException {
            Scanner fileScan = new Scanner(new File(“test.txt”));
            Scanner lineScan;
            String line, word, vowels=“aeiou”;
            int nbVowels=0, nbChars=0, nbWords=0;

            while(fileScan.hasNext()) {
                  line = scan.nextLine();
                  lineScan = new Scanner(line);

                  while(lineScan.hasNext()) {
                        word = lineScan.next();
                        nbChars += word.length();
                        nbWords++;
                        for(int i=0;i<word.length;i++) {
                              if(vowels.indexOf(word.charAt(i))>=0)
                                    nbVowels++;
                        }
                  }
            }

            System.out.println(“Nb of words: ” + nbWords);
            System.out.println(“Ratio: ”+ (double) nbVowels/nbChars);
      }
}
```