COE 211/COE 212 – Computer/Engineering Programming

Welcome to
The Midterm Exam II
Wednesday May 09, 2012

Instructor: Dr. Wissam F. Fawaz
Dr. Georges Sakr

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 **125 minutes** to complete the seven 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** right answer.

1) Consider the following code fragment.

```
String Word = "ereht ih";
for(int i=0; i<Word.length(); i++)
      System.out.print(Word.charAt(i);
```

   a. Run-time error
   b. Syntax error
   c. ereht hi
   d. None of the above

2) What is the order of execution for the following code fragment?

```
1. int x=1, value = 0;
2. switch(value) {
3. case 0:
4.   x++;
5. case 1:
6.   x = x + 2;
7.   break;
8. default:
9.   x = x * 2;}
10. System.out.println(x);
```

   a. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
   b. 1, 2, 5, 6, 7, 10
   c. 1, 2, 8, 9, 10
   d. 1, 2, 3, 4, 5, 6, 7, 10

3) Formal parameters passed to a method are examples of:
   a. Instance variables
   b. Local variables
   c. Static variables
   d. None of the above

4) Consider the program segment presented below.

```
int i = 1;
while(i <= 100) {
      System.out.print(i);
      if(i % 10 == 0)
            System.out.println();
      i++;
}
```

   a. All the numbers between 1 and 100, with a line feed after number 10
   b. All the numbers from 1 to 100 in a grid with 10 rows and 10 columns
   c. All the numbers from 1 to 101, with a line break after every 10^{th} number
   d. All the numbers from 1 to 101 with four values per row

5) Consider the following lines of code.

```
String w1 = "Yes";
String w2 = new String("Yes");
if(w1== w2)
      System.out.println("Yes");
```

```
else
   System.out.println("No");
```

The output of the above-presented code is:

a. Yes
b. No
c. Syntax error
d. None of the above

6) What will be the output of the following code segment?

```
String jumbo = "anarchy";
String shrimp = "rules";
int code=1;
if(code == 1)
if(code > 0)
System.out.print("jumbo");
System.out.print("shrimp");
```

a. anarchy
b. rules
c. anarchyrules
d. None of the above

7) What will be the value of the variable "s" after the following code is executed?

```
int s, p = 2, q = 3, r = 4;
if(p < q) if(r < q)
s = q; else s = r;
else s = p;
s++;
```

a. 4
b. 3
c. 5
d. 2

8) Which of the following expressions will check to see if x is not equal to y:

a. x =! y
b. x < y && x > y
c. ! (x = = y)
d. Both (b) and (c)

9) If the boolean variable `x` is initially `false`, then the following boolean expression `(x != true) && !x || x` yields:

a. false
b. true
c. syntax error
d. run-time error

10) In Lexicographic ordering, which of the following strings: "14all", "14b", "Good", "Goodbye", and "good" would come first?

a. 14all
b. Good
c. good
d. 14b

Problem 2: Long true or false questions (**10 minutes)** [10 points]
In the following questions, check all the correct answers. There is **at least** one correct answer per question, but **there may be more**.

1. Which of the following are **false** about the `switch` statement?
   - **a. The expression included in the header of a switch statement can be of type boolean.**
   - b. It is possible to implement a switch statement using if statements.
   - **c. The body of a switch statement must contain at least one break statement.**

2. Which of the following are **false** regarding constructors?
   - **a. The name of a constructor must be the same as the class name only if it is a public constructor. If it is a private constructor, then you can change its name.**
   - **b. One way to tell the difference between a constructor and a regular method is that a constructor does not return any value.**
   - c. A constructor is a special initialization code that is run when an object is created.

3. Which of the following are **true** about static class members
   - a. A static method cannot access static instance variables.
   - b. PI( ) is a static method defined in the `Math` class.
   - **c. Static variables can be used by a non-static method.**

4. Which of the following are **false** about interfaces?
   - a. `Comparable` is a Java built-in interface that is implemented by the `String` class.
   - **b. The `Iterator` interface defined in the `java.util` package offers two methods, namely the `nextLine()` and `hasNext()` methods.**
   - c. An interface is a collection of abstract methods.

5. Which of the following are **false**?
   - a. Two methods in Java can have the same name and the same number of parameters.
   - b. If `toString()` is not implemented in a class, then printing an object from that class displays the object's class name and address in memory.
   - **c. Both the `do-while` and `for` repetition statements execute their body one or more times.**

6. Which of the following are **true**?
   - a. The `length` variable can be used to determine the number of characters stored in a String.
   - b. An instance variable is shared among all the objects created from a class.
   - **c. When defining a formal parameter, you are required to specify its data type.**

7. Which of the following are **false** about repetition statements?
   - **a. It is possible to create an infinite loop out of `while` and `do-while` loops but not `for` loops.**
   - **b. The following statement is syntactically valid: for(int j=0, j < 100, j++) j--;**
   - c. The following `while` loop is an infinite loop: while(true) i++;

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=0, i;
  for(i=0; i < 5; i++)
       x += i;
  x += i;
```
Answer: x =**15**

(2)
```
int x = 0;
   for(int i=1; i<= 2; i++)
       for(int j=4; j<6; j++)
            x += (i*j);
```
Answer: x =**27**

(3)
```
boolean x=true, y=true, z=false;
  x = !x && y || (z==x);
```
Answer: x =**false**

(4)
```
int x=0, y= 1234, z = 1000;
  do {
       x = x*10 + y/z;
       y = y % z;
       z = z/10;
  } while(z >= 1);
```
Answer: x =**1234**

(5)
```
int count=15, value=3, limit=9, x=0;
  while(count > limit) {
       x = x + value;
       count--;}
```
Answer: x =**18**

(6)
```
String x;
  int y = 0;
  x = “Change is ”+y+((y!=1)?“Dime”:“Dimes”);
```
Answer: x =**Change is 0 Dime**

(7)
```
boolean x; int value1 = 6, value2 = 12;
  x =!(value1 >= value2)&&(value1!=value1-value2);
```
Answer: x =**true**

Problem 4: Short true or false questions (**10 minutes)** [10 points]

1. A code that loops a predetermined number of times is best represented using a **counter** variable with a **while** loop.

Answer: True **False**

2. A program that contains **if-else** statements could be rewritten using **if** statements **only**, that is, without the **else** parts.

Answer: **True** False

3. In Java, every **if** statement must use curly braces { }, otherwise the code will not compile.

Answer: True **False**

4. When placing inside a loop body a **continue** statement immediately followed by a **break** statement, then it is the same as if both of them were not there. In other words, they will cancel each other out.

Answer: True **False**

5. Java methods are **private** by default, which means that when no visibility modifier is applied to a method, it cannot be called from outside of the class that contains it.

Answer: True **False**

6. The output of the following statements is: 10 Done

```
int y = 10, z = 0;
System.out.println("" + z + y + "Done");
```

Answer: True **False**

7. The statement `if(a > b) a++; else b--;` will do the same thing as the statement `if(a < b) b--; else a++;`

Answer: True **False**

8. The string `Hello` gets printed 40 times after the following code is executed:

```
for(int i = 0; i < 8; i+=2)
for (int j=1; j<=10; j++)
System.out.println("Hello");
```

Answer: **True** False

9. The following do-while loop is an infinite loop

```
int x=10;
do{ x*=20 } while(x > 5);
```

Answer: **True** False

10. After the following code has been executed, x will end up storing a value of 40

```
int x=10;
for(int y=5; y<20; y+=5)
     x+=y;'
```

Answer: **True** False

Problem 5: Code analysis (**10 minutes)** [8 points]

1) Consider methodA shown at right.
How would you best describe its return value?

a. x + y
b. x * x
c. x * y
d. x^y

```
int methodA(int x, int y)
{
      int z = 0;
      for(int i=0;i<y;i++)
            z = z + x;
      return z;
}
```

2) Consider methodB shown at right.
How would you best describe its return value?

a. x^y
b. x * x
c. x * y
d. None of the above

```
int methodB(int x, int y)
{
      int z = 1;
      for(int i=0;i<y;i++)
            z=methodA(z,x);
      return z;
}
```

3) Consider methodC shown at right.
What is output of the call: methodC(1324);

a. 4
b. 1
c. 10
d. 3

```
int methodC(int number)
{
      int x = number;
      int count = 0;
      while(x > 0) {
            x = x/10 ;
            count++ ;
      }

      int i=0;
      for(i=0;i<count/2;i++)
        number=number/10;
      System.out.println(
            number%10);
}
```

Problem 6: Method definition (**15 minutes)** [10 points]

You are given below the headers of 4 methods called `isDivisible`, `beginsWithCons`, `countB`, and `gcd`. Your job is to complete the definition of each one of these methods as per the provided guidelines.

1. **isDivisible** is a method that accepts two integer parameters (denoted by **x** and **y**) and returns true if **x** is divisible by **y** and false otherwise.

```
public boolean isDivisible(int x, int y) {


   return (x%y==0);


}
```

2. **beginsWithCons** is a method that accepts a `String` parameter and returns a boolean value that indicates whether the String used as a parameter begins with a consonant.

```
public boolean beginsWithCons(String word) {

String vowels="aeiou";
char c=word.charAt(0);
return (word.indexOf(c)==-1);
}
```

3. **countB** is a method that accepts a `String` parameter and returns the number of times the letter 'B' appears in the String.

```
public boolean countB(String word) {
int cntB=0;
for (int i=0;i<word.length();i++)
      if (word.charAt(i)=='B') cntB++;
return cntB;
}
```

4. **gcd** is a method that accepts 2 `int` parameters and returns their greatest common divisor (gcd).

```
public int gcd(int a, int b) {
while(a!=b)
{
if (a>b)
    a=a-b;
else
    b=a-b;
}
return a;

}
```

Problem 7: Coding (**55 minutes**) [45 points]

1. Design and implement an application that determines and prints the number of odd, even, and zero digits in an integer value read from the end user.
   Sample output:
   Enter an int value: 1024
   Number of zeros in entered value: 1
   Number of odd digits in entered value: 1
   Number of even digits in entered value: 2

```
import java.util.Scanner;


public class CntOddEven {


     public static void main(String[] args) {
          Scanner scan=new Scanner(System.in);
          System.out.print("Enter a Number: ");
          int n=scan.nextInt();
          int cnt0=0;
          int cntOdd=0;
          int cntEven=0;
          while (n>0)
          {
               int lastDigit=n%10;
               n/=10;
               if (lastDigit%2==0)
                    if (lastDigit==0)
                         cnt0++;
                    else
                         cntEven++;
               else
                    cntOdd++;
          }
          System.out.println("Number of 0 in entered value:
"+cnt0);
          System.out.println("Number of even digits in
entered value: "+cntEven);
          System.out.println("Number of odd digits in
entered value: "+cntOdd);

     }
}
```

2. Design and write a Java application that takes as input an integer larger than 1 and prints the sum of the squares from 1 to that integer (inclusive). For example, if the integer equals 4, then the value that should be printed out is 30, which is the sum of squares between 1 and 4 (i.e., 1+4+9+16).

```
import java.util.Scanner;


public class SumSquares {


   public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.print("Enter a Number: ");
        int n=scan.nextInt();
        int sum=0;
        for (int i=1;i<=n;i++)
             sum+=i*i;
        System.out.println(sum);

   }
}
```

3- Design and implement an application that reads a string from the user, then determines and prints how many of each lowercase vowel (a, e, i, o, and u) appears in the entire string.

```
import java.util.Scanner;

public class CountVowels {

	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		System.out.print("Enter a word: ");
		String word=scan.nextLine();
		int cntA=0,cntE=0,cntI=0,cntO=0,cntU=0;
		for (int i=0;i<word.length();i++)
			switch(word.charAt(i))
			{
			case 'a':
				cntA++;
				break;
			case 'e':
				cntE++;
				break;
			case 'i':
				cntI++;
				break;
			case 'o':
				cntO++;
				break;
			case 'u':
				cntU++;
				break;
			}
		System.out.println("Number of a: "+cntA);
		System.out.println("Number of e: "+cntE);
		System.out.println("Number of i: "+cntI);
		System.out.println("Number of o: "+cntO);
		System.out.println("Number of u: "+cntU);

	}

	}
```

4- Write a program that prompts the user to input the x-y coordinates of a point in a Cartesian plane. The program should then output a message indicating whether the point is the origin, is located on the x axis, the y axis, or appears in a particular quadrant. For example (0, 0) is the origin; (4, 0) is on the x-axis; (0, -3) is on the y-axis; (2, 3) is in the first quadrant; (-2, 3) is in the second quadrant, and (-2, -3) is in the third quadrant.

```
import java.util.Scanner;

public class Quadrant {

	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		System.out.print("Enter x: ");
		int x=scan.nextInt();
		System.out.print("Enter y: ");
		int y=scan.nextInt();
		if (x>0 &&y>0)
			System.out.println("1st Quadrant");
		if (x<0 &&y>0)
			System.out.println("2nd Quadrant");
		if (x<0 &&y<0)
			System.out.println("3rd Quadrant");
		if (x>0 &&y<0)
			System.out.println("4th Quadrant");
		if (x==0 &&y!=0)
			System.out.println("On the x axis");
		if (x!=0 &&y==0)
			System.out.println("On the y axis");
		if (x==0 &&y==0)
			System.out.println("Origin");

}

}
```

5- Write a program that reads a positive value N from the user and prints N random values between 0.0 (inclusive) and N (exclusive), and then prints their average value.

```
import java.util.Scanner;


public class RandomNum {


public static void main(String[] args) {
     Scanner scan=new Scanner(System.in);
     System.out.print("Enter a Number: ");
     int n=scan.nextInt();
     double sum=0;
     for (int i=0;i<n;i++)
     {
          double rnd=n*Math.random();
          System.out.println(rnd);
          sum+=rnd;
     }
     System.out.println("The average is: "+sum/n);

}

}
```

6- Write a program that reads a positive value N from the user and prints only the prime number up to N, followed by their sum.

```
import java.util.Scanner;


public class PrimeSum {

	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		System.out.print("Enter a Number: ");
		int n=scan.nextInt();
		int sum=0;
		for (int i=2;i<=n;i++)
		{
			if (isPrime(i))
			{
				System.out.println(i);
				sum+=i;
			}
		}
		System.out.println(sum);

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