COE 211/COE 212 – Computer/Engineering Programming

Welcome to Exam I
Monday November 12, 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 **110 minutes** to complete the required 8 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 (**15 minutes**) [15 points]

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

1) What makes a method to be a **constructor** is the fact that
   - a. it has the same name as the class
   - b. it has a return type of void
   - c. all of the above
   - **d. none of the above**

2) All variables must be given an **initial value** before they are used since **otherwise**
   - **a. the compiler issues an error message**
   - b. the interpreter complains
   - c. a run-time error is produced
   - d. both (b) and (c)

3) Which of the following code fragments causes the **garbage collector** to perform garbage collection?
   - a. String str1 = "Hi";
     String str2 = " there";
     String str = str1 + str2;
   - b. String str, str1 = "Hi";
     String str2 = " there";
     str = str1 + str2;
   - **c. String str1 = "Hi";**
     String str2 = " there";
     str1 += str2;
   - d. None of the above

4) Identify the code fragment that results in a **compile-time error**:
   - **a. double a = 32.5;**
     float b = 45.6;
     a = b;
   - b. double a = 32.5;
     int b = 45;
     a = b;
   - c. long a = 32L;
     int b = 45;
     a = b;
   - d. None of the above

5) Given that a = bx^2 + 5; which of the following is a **valid Java statement** for this equation?
   - a. int a = b * Math.pow(x, 2) + 5;
   - **b. double a = (double) (b * x * x + 5);**
   - c. double a = b (Math.pow(x, 2)) + 5.0;
   - d. None of the above

6) Which of the following modifiers **violates the encapsulation principle** when applied to a variable
   - **a. public**
   - b. static
   - c. private
   - d. None of the above

7) To extract the **first two characters** of the `String` stored in a variable called `str`, you would do:
   a. str.substring(1, 2);
   b. str.substring(0, 1);
   c. str.substring(0, 1) + str.substring(1, 3);
   d. **str.charAt(0) + "" + str.charAt(1);**

8) Which of the following expressions results in a **value of zero**?
   a. 3 / 2
   b. 4 / 2 / 2
   c. **16 % 4 % 4**
   d. None of the above

9) Examine the following code fragments and select the one that leads to a **loss of information**:
   a. double a = 45.6;
   b. **long a = (long) Math.sqrt(18);**
   c. int a = (int) Math.pow(2, 3);
   d. None of the above

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

11) A Java program that uses the `DecimalFormat` class **can import**
   a. java.text.DecimalFormat
   b. java.util.DecimalFormat
   c. java.text.*
   d. **Either (a) or (c)**

12) Which of the following represents the **header** of a `toString` method?
   a. private String toString( )
   b. public String toString(int a)
   c. public toString( )
   d. **None of the above**

13) What **value** will z store if we execute the following assignment statement?
   `double z = (int) Math.random() * 7 + 1;`
   a. z will have a randomly generated value between 1 and 7
   b. z will have a randomly generated value between 0 and 6
   c. **z will equal 1.0**
   d. None of the above

14) The parameters that are used when **defining a method** are referred to as
   a. **formal parameters**
   b. actual parameters
   c. instance parameters
   d. informal parameters

15) Which of the following requests **memory from** the system to store an object
   a. **new**
   b. dot operator
   c. this
   d. None of the above

Problem 2: Long true or false questions (**10 minutes**) [12 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**:
   a. Inside a mutator method, the value of a local variable is assigned to an instance variable.
   b. **The instance variable is used to initialize a local variable inside a mutator method.**
   c. **An accessor method uses `void` as its return type.**

2. Which of the following are **true**:
   a. Empty parentheses following a method name indicates that the method does not produce any output.
   b. You cannot call the static methods of another class until you create an object of that class.
   c. **Constructors cannot include a `return` statement.**

3. Which of the following are **false** about methods:
   a. **Variables declared in a method are instance variables and can be used only in that method.**
   b. A return statement is required at the end of every getter method.
   c. Methods define the behavior of a class.

4. Which of the following are **true** statements:
   a. **It is possible in Java to write a single line of code that generates 2 lines of output.**
   b. A Java program may not give the same output when all indentations are removed.
   c. **It is possible in Java to write multiple lines of code that generate 1 line of output.**

5. Which of the following are **false**:
   a. **When defining a method, the header of the method must end with a semicolon.**
   b. **A logical error occurs when the compiler encounters code violating Java's rules.**
   c. A client Java program must have a main method as a starting point.

6. Which of the following are **true** about variables:
   a. Any variable in Java declared as `final` becomes a Java reserved word.
   b. The type of a variable must match the type of its value since Java is a weakly typed language.
   c. **Primitive variables do not store addresses.**

7. Which of the following are **false** about wrapper classes:
   a. **The `Integer` wrapper class contains a static method called `parseInt` that can be used to convert a `char` into an `int`.**
   b. **The wrapper class `Int` is part of the `java.lang` package.**
   c. Wrapper classes can be used inside a program without an explicit import declaration statement.

8. Which of the following are **false**:
   a. Integer division yields an integer answer in Java.
   b. **If an arithmetic expression contains nested parentheses, the innermost set is evaluated last.**
   c. The following Java statement: Math.pow(2, 3); would yield a double value of 8.0.

Problem 3: Short true or false questions (**10 minutes**) [12 points]

1. The output of the following statement is: 3
```
System.out.println(2 * (3 / 2));
```
Answer: True   **False**

2. The following are all invalid variable names: `56, static, class, 4all`.
Answer: **True**   False

3. By convention, method names in Java begin with an uppercase first letter.
Answer: True   **False**

4. A primitive-type variable can be used to invoke a method.
Answer: True   **False**

5. Floating point values that appear in a Java program are known as floating-point literals and are of type `float` by default.
Answer: True   **False**

6. The statement given below is a valid Java statement.
```
int If = 7;
```
Answer: **True**   False

7. The output of the statement below is: 1
```
System.out.print( -7 % -3 );
```
Answer: True   **False**

8. Variables declared with private visibility are accessible only to the methods of the class in which they are declared.
Answer: **True**   False

9. If the called method and the calling method belong to two different classes, then only the method name is needed to invoke the called method.
Answer: True   **False**

10. The output of the code shown below is: 15 Done
```
System.out.print( "" + 10 + 5 + " Done");
```
Answer: True   **False**

11. If `String a = "horse";` and `String b = "horsefly";` then `a.compareTo(b);` returns a negative value.
Answer: **True**   False

12. A variable of type `String` can be used to hold a single character.
Answer: **True**   False

Problem 4: Evaluating java expressions (**15 minutes**) [7 points]
For each of the following code fragments, what is the value of x after the
statements are executed?

(1)
```
String x = "\"seven\" equals: ";
x += "seventy".length();
```
Answer: x = *"seven" equals: 7*

(2)
```
String str = "hi5.05";
String last = str.substring(2, str.length()-1);
double val = Double.parseDouble(last);
double x = Math.ceil(val);
```
Answer: x = *5.0*

(3)
```
int y = 3, z = 7;
x = change(y, z);

public int change(int a, int b) {
    int temp = b; b = a; a = temp;
     return a - b;
}
```
Answer: x = *4*

(4)
```
int y = 12;
int z = y++;
int x = ++(z - y);
```
Answer: x = *0*

(5)
```
String sentence = "Exam Fun";
String str1 = sentence.substring(5, 8);
 String str2 = sentence.substring(0, 4);
String x = str1.concat(" ").concat(str2);
```
  Answer: x =*Fun Exam*

(6)
```
String number = "25,67"; String one = "1";
String str1 = number.replace(',', '.');
String str2=str1.substring(2, str1.length());
 str2 = one.concat(str2);
double x = Double.parseDouble(str2) -
           Double.parseDouble(str1);
```
   Answer: x =-*24*

(7)
```
DecimalFormat fmt = new DecimalFormat("00.##");
String str1 = fmt.format("2.017");
str1 = str1.replace('0', '2');
String x = str1.replace('2', '0');
```
Answer: x =*00.00*

Problem 5: Object-oriented code analysis (**17 minutes**) [12 points]
Consider the following class:

```
public class Foo{
  private int t, y, m, d;
  public Foo(int t){ this.t=t; y=this.t/360;
                     d=this.t%360; m=d/30; d=d%30; }
  public int getY() { return y; }
  public int getM() {return m; }
  public int getD() { return d; }
  public int sum(int d) { return t + d; }
  public String toString() {
  String output = "Y: "+getY()+",M: "+m+",D: "+getD();
  return output; }}
```

This class is supplemented with the following driver class:

```
public class Driver{
  public static void main(String[] args) {
     Foo f1 = new Foo(392);
     Foo f2 = new Foo(f1.sum(28));
     int i=f1.toString().indexOf(":");
     String str = f1.toString().substring(i+2, i+3); }}
```

For each of the following statements, indicate the expression that will be
printed out. Display the output in the provided boxes.

a) `System.out.println(f2);`

Y: 0,M: 1,D: 2

b) `System.out.println(i);`

1

c) `System.out.println(str);`

1

d) `int d1 = f1.getM()*30 + f1.getD();`
   `int d2 = f2.getM()*30 + f2.getD();`
   `System.out.print("Diff:"+Math.abs(d1-d2));`

Diff: 0

e) `String s = f2.toString();`
   `int index = s.indexOf(",");`
   `String mutation =`
        `s.substring(index+1, index+6);`
   `System.out.println(mutation);`

M: 1,

f) `System.out.println(`
   `f1.toString().charAt(str.length()+i));`

White space

Problem 6: Code Analysis (**10 minutes**) [10 points]

1) What is the output of the code given in the two columns below when an object of type `Confuse` is created and used to call the method called `startUp()`?

```
public class Confuse{
    private int x;
    private int y;
    public Confuse(){
        x=2; y=3;
    }

    private void first(int a){
        x = a; x++;
    }

  private void second(int s,int t){
        setXY((y+s), (x-t));
    }
}
```

```
private void setXY(int s,int t){
        x = s;
        y = t;
    }

    private void display(){
        System.out.println(x+y);
    }

    public void startUp() {
        first(x);
        second(y, x);
        display();
    }}
```

    a. 1
    b. **6**
    c. 10
    d. None of the above

2) Consider the code given below. If its output is: **Date: Date@F8762**. What is the best explanation?

```
Date startingDate = new Date();
System.out.println("Date: " + startingDate);
```

    a. The default constructor was not written properly
    b. Some fields were not initialized, so a random value was printed
    c. **The `toString()` method was not created for the class Date**
    d. None of the above

Problem 7: Fill in the blanks (**8 minutes**) [12 points]

Fill in the blanks in the following class definition:

```
public class Blanks
{
     private_ int member;


     public void_ setMember (_int_ member)
     {
          _this.member__ = member;
     }

     __public__ _int_ getMember(_____)
     {
          _return__ member;
     }

// Constructor: sets member field to the given
//              parameter value.

     public __Blanks__ (int value)
     {
          ___member = value___;
     }

// The method below gets invoked whenever
// you attempt to print a Blanks object.
// It should return the instance variable as a string.

     public __String toString ()
     {
          _____return member + "";_____;
     }

}
```

Problem 8: Coding (**25 minutes**) [20 points]

1. Design and implement a class called `UnitAverage` that reads 2 integers from the user and prints the average of their unit numbers. Note that the unit number is nothing else but the digit stored at the one's position in an integer value. So, for example, if the user enters 65 and 94, your program must print the average of their unit numbers (5 and 4 respectively) which is 4.5.

Sample run:
Enter the first number: 45
Enter the second number: 62
The average of 5 and 2 is: 3.5

```java
import java.util.Scanner;


public class UnitAverage {


    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int a=s.nextInt();
        System.out.print("Enter the second number: ");
        int b=s.nextInt();
        int unit1=a%10;
        int unit2=b%10;
        double av=(unit1+unit2)/2.0;
        System.out.println("The average of "+unit1+"
and "+unit2+" is :"+av);


    }

}
```

2. The electrical resistance of a wire is given by:

$$R = P\frac{L}{A}$$

Where P is the resistivity of the wire, L is its length, and A is its area. Write a program that reads the values of **P** and **L** from the user as well as the diameter **d** of the wire and then outputs the resistance **R** of the wire. The area of the wire is given by $A = \pi d^2/4$ and as such the area can be computed based on **d**, the input diameter value. Make sure that you format the output resistance value **R** to 2 decimal places.

Sample run:
Enter P: 1.257736
Enter L: 10.5
Enter d: 0.008
The resistance of the wire is: 262729.56 ohm

```java
import java.text.DecimalFormat;
import java.util.Scanner;
public class Resistance {

    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter P: ");
        double p=s.nextDouble();
        System.out.print("Enter L: ");
        double l=s.nextDouble();
        System.out.print("Enter d: ");
        double d=s.nextDouble();
        double area=Math.PI*d*d/4;
        double r=p*l/area;
        DecimalFormat fmt=new DecimalFormat("0.##");
        System.out.println("The resistance of the wire
is: "+fmt.format(r)+" ohm");

    }

}
```