

# COE 212 – Engineering Programming

Welcome to Exam I  
Tuesday October 24, 2017

Instructors: Dr. Dima El-khalil  
Dr. Jihad Fahs  
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 **115 minutes** to complete **6** the 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 (20 minutes) [13 points]

For the questions given below, consider the following driver and helper classes:

```
public class MultiAccts {
public static void main(String[] args){
    FacebookAcct acct1 =
new FacebookAcct("Ledeg", "Deg123");
    FacebookAcct acct2 =
new FacebookAcct("Messi", "Barca");
    FacebookAcct acct3 =
new FacebookAcct("Ronaldo", "RealE");
    acct2.login();
    acct3.login();
    acct1.setPassword(acct1.getPassword()+
acct1.getUserName().charAt(0));
    acct2.setPassword(
acct2.getPassword().substring(2,5));
    acct3.setPassword(
acct3.getPassword().replace('e', 'x'));
    System.out.println(acct1);
    System.out.println(acct2);
    System.out.println(acct3);  }}
```

```
public class FacebookAcct {
private String userName, password;
private int loginAttempts;
public FacebookAcct(String u, String p){
    userName=u; password=p; loginAttempts=0;}
public int getAttempts() {
    return loginAttempts;}
public String getUserName(){
    return userName;}
public String getPassword(){
    return password;}
public void login(){incrementAttempts();}
private void incrementAttempts(){
    loginAttempts+=1;}
public void setPassword(String p){
    password = p;}
public String toString(){
    return userName + ":" + password +
        ":" + loginAttempts;  }}
```

- 1) What output does the first `println` statement of the driver class produce?
  - a. Ledeg:Deg123:1
  - b. Ledeg:Deg123L:1
  - c. Ledeg:Deg123l:1
  - d. None of the above**
- 2) What output does the second `println` statement of the driver class produce?
  - a. Messi:rc:1
  - b. Messi:arc:1
  - c. Messi:rca:1**
  - d. None of the above
- 3) What output does the third `println` statement of the driver class produce?
  - a. Ronaldo:Rxalx:1
  - b. Ronaldo:RxalE:1**
  - c. Ronaldo:RxalX:1
  - d. None of the above
- 4) How many mutator methods does the helper class contain?
  - a. 0
  - b. 1**
  - c. 2
  - d. None of the above
- 5) How many methods are defined inside the driver class?
  - a. 1**
  - b. 2
  - c. 3
  - d. None of the above
- 6) Consider the `acct1` object created in the driver class. Which of the following can be used to correctly extract the last character of its associated user name?
  - a. `acct1.getUserName().charAt(length()-1)`
  - b. `acct1.getUserName.charAt(acct1.length())`
  - c. `acct1.getUserName().substring(acct1.getUserName().length()-1,acct1.getUserName().length()-1)`**
  - d. Both (b) and (c)

- 7) Which of the following statements results in a compile time error when inserted into the body of the main method of the driver class? Assume that the statements are placed after the instantiation statement of the acct2 object.
- acct2.incrementAttempts();**
  - char letter = acct2.getPassword().charAt(acct2.length());
  - Both of the above
  - None of the above
- 8) Consider the acct3 object created in the driver class. Which of the following statements correctly extracts the number of login attempts from the String representation of acct3 and then stores it in an int variable called attempts?
- int attempts = Integer.parseInt(acct3.charAt(acct3.length()-1));
  - int attempts = Integer.parseInt(acct3.toString().charAt(acct3.length()-1));
  - int attempts = Integer.parseInt(acct3.toString().substring(acct3.toString().length()-1,acct3.toString().length()));**
  - Both (b) and (c)
- 9) Which of the following statement correctly sums up the number of login attempts associated with acct2 and acct3 before storing the resulting sum in a String variable called sum?
- String sum=acct2.getAttempts()+acct3.getAttempts();
  - String sum= ""+acct3.getAttempts()+acct2.getAttempts();
  - String sum=acct2.getAttempts()+acct1.getAttempts()+"";
  - None of the above**
- 10) Assuming that the sum variable from the previous question has been created correctly, which of the following can be used to correctly extract its value and then store it in a variable called value?
- double value=Integer.parseInt(sum);
  - int value=Integer.parseInt(sum);
  - Both of the above**
  - None of the above
- 11) Which of the following statements correctly determines the length of the password associated with acct1 and then stores it in an int variable called len?
- int len = acct1.getPassword().length ;
  - int len = acct1.toString().length();
  - int len = acct1.getPassword.length();
  - None of the above**
- 12) Which of the following statements correctly computes the square root of the number of login attempts associated with acct2 and then stores it in a variable called squareRoot?
- int squareRoot = Math.sqrt(acct2.getAttempts());
  - double squareRoot = Math.sqrt(acct2.getAttempts());**
  - Both of the above
  - None of the above
- 13) Which of the following correctly divides the value of  $\pi$  by the number of attempts associated with acct3 and then stores the resulting value in a variable called root?
- Double root = Math.PI/acct3.getAttempts();
  - double root = Math.PI()/acct3.getAttempts();
  - double root = Math.PI/acct3.getAttempts();
  - Both (a) and (c)**

**Problem 2: True or false questions (10 minutes) [12 points]**

1. The following Java statement results in a compile time error:  

```
System.out.println("///");
```

 Answer: True **False**
  
2. The following method definition results in a compile time error:  

```
private void foo() {int output=3; return output;}
```

 Answer: **True** False
  
3. The following code fragment outputs: 23.46  

```
DecimalFormat fmt = new DecimalFormat("0.##");
double value = fmt.format("23.457");
System.out.print(value);
```

 Answer: True **False**
  
4. Both the static `random()` method of the `Math` class and the static `nextFloat()` method of the `Random` class generate a random floating point value between 0 (inclusive) and 1 (exclusive).  
 Answer: True **False**
  
5. In the presence of a logical error in your Java program, your code compiles correctly and executes without any complaint from the interpreter.  
 Answer: **True** False
  
6. The following statement correctly selects a random character from a `String` object called `str`:  

```
char rndChar = str.charAt((int) (Math.random()*str.length()));
```

 Answer: **True** False
  
7. A mutator method provides read only access to one of the instance variables of a class.  
 Answer: True **False**
  
8. The following code fragment increments the variable called `val` by 3.  

```
val++; ++val; val += 1;
```

 Answer: **True** False
  
9. Unboxing is the automatic conversion from a wrapper class to its corresponding primitive data type.  
 Answer: **True** False
  
10. The following statement outputs: 2.0  

```
System.out.print(Math.pow(4, 1/2));
```

 Answer: True **False**
  
11. The following Java statement is valid: `int true = 3;`  
 Answer: True **False**
  
12. The following Java statement is valid: `Boolean flag = false;`  
 Answer: **True** False

**Problem 3: Code analysis (15 minutes) [10 points]**

1) Consider the helper class given below, along with a driver class for it.

<pre>public class ClassA {     private String str;     private int val;     public ClassA(String s, int v) {         initialize(s, v);     }     private void initialize(String s, int v){         str = first(s);         val = second(v);     }     private String first(String s) {         return s.substring(1, s.length()-1);     }     private int second(int v) {         return 2*v;     }     public char third() {         return str.charAt(val);     } }</pre>	<pre>public class ClassADriver {     public static void main(String[] args) {         ClassA obj1 = new ClassA("knife", 1);         char letter1 = obj1.third();          ClassA obj2 = new ClassA("pun", 0);         char letter2 = obj2.third();          ClassA obj3 = new ClassA("permanent", 2);         char letter3 = obj3.third();          System.out.print("" + letter1 + letter2 +             letter3);     } }</pre>
---	---

When running the ClassADriver class, what output is produced?

- fun
- iua
- An error occurs during the execution of the code
- It doesn't compile correctly
- None of the above

2) Consider the class given below, along with a driver class for it.

<pre>import java.text.DecimalFormat; public class ClassB {     private char letter;     private double val;     private DecimalFormat fmt;     public ClassB(char c) {         letter = c;         val = 45.678;         fmt = new DecimalFormat("0.##");     }     private String first(double v, char c){         return c + fmt.format(v);     }     public String toString() {         return first(letter, val);     } }</pre>	<pre>public class ClassBDriver {     public static void main(String[] args){         ClassB obj = new ClassB('1');         String obj_str = obj + "";          System.out.print(obj_str);     } }</pre>
---	---

When running ClassBDriver class, what output is produced?

- 45.68
- 145.68
- An error occurs during the execution of the code
- It doesn't compile correctly**
- None of the above

**Problem 4: Code analysis (15 minutes) [10 points]**

Consider the following class:

```
class Blueprint {
    private int x, y, z;
    public Blueprint(int a, int b){
        x=a; y=a/2; z = a + x; }
    public int getx() { return x; }
    public int gety() { return y; }
    public int getz() { return z; }
    public int method1() { return x + y * z; }
    public String toString() {
        return "x=" + x + ", y="+y+", z=" + z; }
}
```

This class is supplemented with the following driver class:

```
public class Driver{
    public static void main(String[] args) {
        Blueprint B1 = new Blueprint(2, 3);
        Blueprint B2 = new Blueprint (B1.method1(), B1.gety());
        int i = B1.toString().indexOf("+");
        String S = B2.toString().substring(i+1);
    }
}
```

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

1. `System.out.println(B1);`

**x=2, y=1, z=4**

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

**-1**

3. `System.out.println(S);`

**x=6, y=3 z=12**

4. `int x = B1.getx()*10 + B1.getz();`  
`int y = B2.gety()*10;`  
`System.out.println("Answer=" + Math.pow(x - y, 2));`

**Answer=36.0**

5. `System.out.println(B1.toString().charAt(S.length()/2));`

**=**

**Problem 5: Evaluating Java Expressions (15 minutes) [15 points]**

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

```
(1) int x = (int) Math.random()*3 - 1;
    int y = ++x; x = y++;
```

**Answer: x= 0**

```
(2) double val1 = 11.4455;
    double val2 = Math.floor(val1*100);
    double x = val2 + (int) val1*10;
```

**Answer: x= 1254.0**

```
(3) String S1 = new String("Work more");
    String S2 = new String("Expect less");
    int x = S1.toUpperCase().substring(3).indexOf(S1.charAt(3));
```

**Answer: x= -1**

```
(4) DecimalFormat fmt = new DecimalFormat("###.000");
    String str = fmt.format(2.2222).substring(0,2);
    char x = str.charAt(str.length()-1);
```

**Answer: x= '.'**

```
(5) String str1 = "Scientist", str2 = new String("Science");
    str2 = str2.substring(0, 5)+ str1.substring(5);
    String x = str1.equals(str2) + "";
```

**Answer: x= "true"**

```
(6) DecimalFormat fmt1 = new DecimalFormat("0.###");
    DecimalFormat fmt2 = new DecimalFormat("0.#");
    double val = Double.parseDouble(fmt1.format(1.1111));
    char x = fmt2.format(val).charAt(1);
```

**Answer: x= '.'**

```
(7) String str = "54321";
    boolean x = str.equals('5'+4+"3"+"21");
```

**Answer: x= false**

```
(8) String str = "String str = new String()";
    String x = str.substring(str.length() - str.substring(1,
    3).length());
```

**Answer: x= "()"**

```
(9) String S = new String("/////");
    String x = (S.length() + "" + 2 + 2);
```

**Answer: x= "622"**

```
(10) int u = 2, v =3, w = 4;
    u = v--; v = --w; w = u;
    int x = u + v + w ;
```

**Answer: x= 9**

**Problem 6: Coding Problems (40 minutes) [40 points]**

1. Write a program called `NameInQuotes` which reads from the user a string name that contains the user's first name followed by her/his last name with a white space character in between. The program should create and then print-out on-screen a new string `nameVar` consisting of the first name between double quotes followed by the last name between double quotes, with a white space character in between.

**Sample run:****Enter your name: Karl Sagan****The resulting name: "Karl" "Sagan"**

```
import java.util.Scanner;

class NameInQuotes{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter your name: ");
        String name = scan.nextLine();

        int i = name.indexOf(' ');

        String nameVar = "\"" + name.substring(0, i) + "\"" + " " + "\"" +
            name.substring(i+1) + "\"";

        System.out.print("The resulting name: " + nameVar);
    }
}
```



2. Write a Java program called `RandomCosine` that generated a random angle value between  $0^\circ$  and  $360^\circ$  and then computes its cosine.

**Sample run:**

**Random angle: 180**

**Cosine(180)=-1.0**

```
import java.util.Random;

class RandomCosine{

    public static void main(String[] args)
    {

        Random rnd = new Random();

        int x = rnd.nextInt(361);
        System.out.println("Random angle: " + x);

        double y = Math.cos((x * Math.PI)/180);
        System.out.println("Cosine(" + x + ")=" + y);

    }

}
```

3. Write a program called `MiddleChar` which reads from the user a `String` object `S1` consisting of an odd number of characters, and then generates a `char` variable `c` containing `S1`'s middle character and a new `String` object `S2` which consists of all the characters of `S1` except for its middle character.

**Sample run:**

**Enter string S1: Compute**

**Middle character: p**

**String S2: Comute**

```
import java.util.Scanner;

class MiddleChar{

    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter string S1: ");
        String S1 = scan.nextLine();

        int i = S1.length()/2;
        char c = S1.charAt(i);
        System.out.println("Middle character: " + c);

        String S2 = S1.substring(0, i) + S1.substring(i+1);
        System.out.println("String S2: " + S2);
    }
}
```

4. Write a program called `LebaneseMobileNumber` that randomly generates a Lebanese mobile phone number: starting with either: 70 or 71, followed by a dash symbol "-", followed by 6 digits where the 1<sup>st</sup> digit can neither be 0 nor 1. The program then prints the number on-screen.

For instance, 74-224224, 03-012213, 70-131435 are not valid Lebanese mobile numbers since the first phone number starts with 74, the second phone number contains a 0 after the dash, and the third phone number contains a 1 after the dash. However, 71-224224, 03-312213, and 70-231435 are perfectly valid numbers.

```
import java.util.Random;
import java.text.DecimalFormat;

class LebaneseMobileNumber{

    public static void main(String[] args)
    {

        Random rnd = new Random();

        String number = "7";

        int x = rnd.nextInt(2);
        number += x + "-";

        int y = rnd.nextInt(8) + 2;
        number += y;

        int z = rnd.nextInt(100000);

        DecimalFormat fmt = new DecimalFormat("00000");
        number += fmt.format(z);

        System.out.print("A random lebanese number: " + number);

    }
}
```