

COE 212 – Engineering Programming

Welcome to Exam I
Thursday October 22, 2015

Instructors: Dr. Bachir Habib
Dr. Salim Haddad
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 **110 minutes** to complete the required **5** 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 (25 minutes) [14 points]

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

```
public class Employee{
    private String fullName;
    private double salary;
    public Employee(String f,double s){fullName=f; salary=s;}
    public String getFullName(){return fullName;}
    public double getSalary(){return salary;}
    public void setSalary(double s) {salary = s;}
    public void setFullName(String f) {fullName = f;}
    public String toString() {
        return "FullName: "+fullName+", salary: " + salary;}}
public class EmployeeDetails{
    public static void main(String[] args) {
        Employee e1 = new Employee("Wissam Fawaz",2500.0);
        Employee e2 = new Employee("Joe Tekli", 2000.0);
        System.out.println(e1.getFullName());
        System.out.print(e2.getSalary());}}
```

- 1) What output does the **driver class** produce when executed?
 - a. Wissam Fawaz
2000
 - b. Joe Tekli
2000
 - c. Joe Tekli
2500
 - d. None of the above**
- 2) Which of the following can be used to extract the **first name** of the **second employee** e2?
 - a. e2.getFullName().substring(0, 4);
 - b. e2.toString().substring(10, 12);
 - c. e2.getFullName().substring(4);
 - d. None of the above**
- 3) Which of the following can be used to extract the **last name** of the **first employee** e1?
 - a. e2.toString().substring(17, 23);
 - b. e2.getFullName().substring(7);
 - c. Both of the above
 - d. None of the above**
- 4) How many **local variables** are used inside the EmployeeDetails class?
 - a. 4
 - b. 5
 - c. 6
 - d. None of the above**
- 5) How many **formal parameters** are used inside the Employee class?
 - a. 0
 - b. 2
 - c. 4**
 - d. None of the above
- 6) How many **instance variables** are used inside the EmployeeDetails class?
 - a. 0**
 - b. 1
 - c. 2
 - d. None of the above

- 7) What value does the following method call **return**?
- ```
e1.toString().charAt(e1.getFullName().length() -
e2.getFullName().length() + 1);
```
- :
  - u
  - N**
  - None of the above
- 8) What **output** does the following code fragment produce assuming that the `DecimalFormat` class was properly imported?
- ```
String str = e1.getSalary() + "";
str = str.concat("56");
DecimalFormat fmt = new DecimalFormat("0.#");
System.out.print(fmt.format(Double.parseDouble(str)));
```
- 2500.6
 - 2500.1**
 - 2500.0
 - None of the above
- 9) Which of the following statements produces a **compile-time error** when used inside the **main** method of the driver class?
- `Integer salary1 = (int) e1.getSalary();`
 - `Double salary1 = e1.salary;`**
 - Both of the above
 - None of the above
- 10) Which of the following statements results in a **compile-time error**?
- `boolean flag = e1.getSalary().equals(e2.getSalary());`
 - `char flag = e1.getFullName().equals(e2.getFullName());`
 - Both of the above**
 - None of the above
- 11) What **output** does the following code fragment produce?
- ```
String st1 = e1.getSalary() + "", st2 = e2.getSalary() + "";
st2 = st2.replace('0', '5');
System.out.print(Double.parseDouble(st2) - Double.parseDouble(st1));
```
- 0.0
  - 55.0
  - 55.5**
  - None of the above
- 12) Consider the following statements:
- ```
e1.setFullName("Joe Tekli"); e1.setSalary(2000.0);
```
- Which of the following is **true** after these statements are executed?
- `e1.toString().compareTo(e2.toString())` returns a value of 0**
 - `e1` and `e2` are aliases of each other
 - Both of the above
 - None of the above is true
- 13) Which of the following statements **correctly increments** the salary of `e1` by 100?
- `e1.getSalary() += 100;`
 - `e1.setSalary(e1.getSalary()+100);`**
 - Both of the above
 - None of the above
- 14) Which of the following correctly computes the **square root** of the salary of `e2`?
- `Math.sqrt(e2.getSalary());`**
 - `Math.pow(e2.getSalary(), 1/2);`
 - All of the above
 - None of the above

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

1. In a Java program, a constructor method can either have a return type of `void` or have no return type listed in the header at all.

Answer: True **False**

2. In a Java program, each opening curly brace `{` must have a corresponding closing curly brace `}` otherwise a run-time error occurs.

Answer: True **False**

3. In Java, for each primitive data type there is a corresponding Wrapper class, including the `boolean` data type. Therefore, the following Java assignment statement is a valid one:
`Boolean flag = true;`

Answer: **True** False

4. The following code fragment outputs a value of `-1`

```
int x, y = 11;
x = ++y ; x -= y ; System.out.print(x);
```

Answer: True **False**

5. If during the execution of a Java program the last reference to a primitive value is lost, the primitive value is marked as garbage and is automatically removed from memory using Java's automatic garbage collection feature.

Answer: True **False**

6. A `String` object can be changed after instantiation since `String` objects are immutable.

Answer: True **False**

7. A method invocation statement may cause the compiler to generate a compile time error even if the number of actual parameters matches exactly the number of formal parameters.

Answer: **True** False

8. A static method such as the `parseDouble` method of the `Integer` wrapper class is called through the name of the class.

Answer: True **False**

9. The `format` method of the `DecimalFormat` class returns a double value.

Answer: True **False**

10. A method whose sole purpose is to help another method do its job can be declared as `private`.

Answer: **True** False

11. A mutator method allows a client program to change the state of an object created from the helper class.

Answer: **True** False

12. The following Java statement is valid: `char Char = 'L';`

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 int x; public ClassA() {x = 1;} public void first(int a) { x = ++a; } public void second() { first(x); x++; } public void third() { second(); ++x; } public int getX(){return x;} }</pre>	<pre>public class ClassADriver { public static void main(String[] args){ ClassA obj = new ClassA(); obj.first(1); obj.second(); obj.third(); double a = (int) Math.random()*obj.getX() + obj.getX(); System.out.print("Answer is: " + a); } }</pre>
--	--

When running the ClassADriver class, what output is produced?

- a. Answer is: 6.0
- b. Answer is: 7
- c. Answer is: 1
- d. It doesn't compile correctly
- e. None of the above**

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

<pre>public class ClassB { private int x=6, y=2; public int first(int a){ y=a+1; return y+1;} public void second(int a,int b) { y = first(a); y=b+1;} public void doIt() { int val = first(y) ; second(y, val) ; System.out.print(x+y);} }</pre>	<pre>public class ClassBDriver { public static void main(String[] args){ ClassB b=new ClassB(); b.doIt(); } }</pre>
--	---

When running ClassCDriver class, what output is produced?

- a. 7
- b. 9
- c. 11**
- d. It doesn't compile correctly
- e. None of the above

Problem 4: Evaluating Java Expressions (20 minutes) [24 points]

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

(1) `double u, v; u = Math.cos(45); v = Math.sin(45);
int x = (int)(u/v) ;`

Answer: x= 0

(2) `int x = (int) Math.random()*2 + 1;
int y = ++x; x = y++;`

Answer: x= 2

(3) `String S1 = new String("conspiracy is easy to understand");
String S2 = new String("complexity is not");
int x =
S1.concat(S2).toUpperCase().substring(8).indexOf(S1.charAt(4));`

Answer: x= -1

(4) `double a = Math.pow(3, 2);
double b = Math.pow(4, 2);
double x = Math.pow(a + b, 1/2);`

Answer: x= 1.0

(5) `String S = new String("\\\\\\\\");
String x = (S.length() + "" + (1 + 1));`

Answer: x= "32"

(6) `String will = "you "; String No = "Sure ";
String I = "with "; String Sure = "I agree ";
String x = No + Sure + I + will + "!";`

Answer: x= "Sure I agree with you !"

(7) `DecimalFormat fmt = new DecimalFormat("000.#");
String str = fmt.format(5.83).substring(0, 3);
char x = str.charAt(str.length()-1);`

Answer: x= '5'

(8) `String str1 = "Opelia", str2 = new String("Opel");
str2 = str2.concat(str1.substring(4, 5));
String x = str1.equals(str2) + "";`

Answer: x= "false"

(9) `DecimalFormat fmt1 = new DecimalFormat("0.##");
DecimalFormat fmt2 = new DecimalFormat("0.#");
double val = Double.parseDouble(fmt1.format(8.876));
char x = fmt2.format(val).charAt(2);`

Answer: x= '9'

- (10) `String str = "12345";`
`boolean x = str.equals("1"+"2'+3+"45");`
Answer: x= true
- (11) `String str = "String str = new String()";`
`char x = str.charAt(str.length() - str.substring(6, 8).length());`
Answer: x= '('
- (12) `int u = 1, v =2, w = 3;`
`u = v++; v = --w; w = u;`
`int x = u + v + w ;`
Answer: x= 6
- (13) `double val1 = 12.3456;`
`double val2 = Math.floor(12.3456*100);`
`double x = val2 - (int) val1*100;`
Answer: x= 34.0
- (14) `String b ="java at 5pm is fun";`
`String x;`
`x=b.replace('j','J').substring(`
`b.length()-3,b.length()-1).toUpperCase();`
Answer: x= "FU"

Problem 5: Coding (40 minutes) [40 points]

1. The polar coordinates of a point in a 2-dimensional plane r and θ can be converted into their Cartesian counterparts x and y using the following formulas:

$$x = r \cos \theta$$

$$y = r \sin \theta$$

Write a Java program called `PolarToCartesian` that reads the values of r and θ (**in degrees**) from the user and then prints out the x and y coordinates of the point.

Sample run:**Enter r: 1****Enter theta (in degrees): 90****The Cartesian coordinates are: x = 0.0 and y = 1.0**

```
import java.util.Scanner;

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

        double r, theta, thetaR, x, y;

        System.out.println("Enter r: ");
        r = scan.nextDouble();

        System.out.println("Enter theta (in degrees): ");
        theta = scan.nextDouble();

        thetaR = (theta * Math.PI)/180;

        x = r * Math.cos(thetaR);
        y = r * Math.sin(thetaR);

        System.out.println("The Cartersian coordinates are: x = " + x + " y = " + y);

    }
}
```

2. Write a Java program called `VectorCosine` that reads from the user two integers representing the X and Y coordinates of a vector V_1 in a 2D space, and then reads two integers representing the X and Y coordinates of another vector V_2 . Your program should then compute the cosine of the angle between them, and print it on-screen rounded to 2 decimal places.

$$\cos(\vec{V}_1, \vec{V}_2) = \frac{X_1 \times X_2 + Y_1 \times Y_2}{\sqrt{((X_1)^2 + (Y_1)^2) \times ((X_2)^2 + (Y_2)^2)}}$$

Sample run:

Provide XY coordinates of vector V1: 2 2

Provide XY coordinates of vector V2: 4 4

The Cosine of the angle between V1 and V2: 1

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

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

        int X1, X2, Y1, Y2;
        System.out.println("Provide XY coordinates of vector 1: ");
        X1 = scan.nextInt(); Y1= scan.nextInt();

        System.out.println("Provide XY coordinates of vector 2: ");
        X2 = scan.nextInt(); Y2= scan.nextInt();

        double cosine = (X1 * X2 + Y1 * Y2) / (Math.sqrt(Math.pow(X1, 2) +
        Math.pow(Y1, 2)) * Math.sqrt(Math.pow(X2, 2) + Math.pow(Y2, 2)));

        DecimalFormat fmt = new DecimalFormat("#.##");

        System.out.println("The cosine of the angle between V1 and V2: " +
        fmt.format(cosine));
    }
}
```

3. Design and implement a Java program called `StringEncoding` that reads a `String` of **four characters** from the user, called `S1`, and then creates a new `String` called `S2`, made of the same four characters of `S1` but randomly reshuffled. Then, the program must print `S2` to the screen.

`S2` can be created by doing the following four times:

- a. Select at random one of the characters from `S1` then
- b. Remove that selected character from `S1`

Sample run:

Enter a String: name

Randomly generated String is: mena

```
import java.util.Scanner;
import java.util.Random;

class StringEncoding {

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

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

        int x = gen.nextInt(S1.length());
        char c1= S1.charAt(x);
        S1 = S1.substring(0, x) + S1.substring(x+1, S1.length());
        System.out.println(S1);

        x = gen.nextInt(S1.length());
        char c2= S1.charAt(x);
        S1 = S1.substring(0, x) + S1.substring(x+1, S1.length());
        System.out.println(S1);

        x = gen.nextInt(S1.length());
        char c3= S1.charAt(x);
        S1 = S1.substring(0, x) + S1.substring(x+1, S1.length());
        System.out.println(S1);

        x = gen.nextInt(S1.length());
        char c4= S1.charAt(x);
        S1 = S1.substring(0, x) + S1.substring(x+1, S1.length());
        System.out.println(S1);

        String S2 = ""+ c1 + c2 + c3 + c4;

        System.out.println("The new string = " + S2);

    }
}
```

4. The quality of a search engine (like *Google Search*) can be experimentally evaluated using two measures: *Precision (PR)* highlighting the percentage of *relevant* documents correctly *retrieved* by the system with respect to all documents *retrieved* by the system, and *recall (R)* highlighting the percentage of *relevant* documents correctly *retrieved* by the system with respect to all documents *relevant* for the human user. High *precision* denotes that the search engine achieved high accuracy, whereas high *recall* means that very few relevant documents were not retrieved by the engine. In addition to comparing one approach's *precision* improvement to another's *recall*, it is a common practice to consider the *F-value* measure, which represents the harmonic mean of *precision* and *recall*:

$$F\text{-value} = \frac{2 \times PR \times R}{PR + R} \in [0,1]$$

Write a Java program called `FValue` that randomly generates two double numbers `PR` and `R`: each in the range of 0 (inclusive) to 1 (inclusive), representing respectively the *precision* and *recall* levels of a certain search engine, and computes the corresponding *f-value* score using the formula above. The program then prints *precision*, *recall*, and *f-value* scores to the screen, rounded to 3 decimal places each.

Sample run:

Given: Precision = 0.810 and Recall = 0.700

F-value = 0.751

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

public class f_value
{
    public static void main (String[] args)
    {
        Random rand = new Random();
        DecimalFormat fmt = new DecimalFormat("0.000");

        double PR = rand.nextInt(101)/100.0;
        double R = rand.nextInt(101)/100.0;
        double F = (2 * PR * R)/(PR + R);

        System.out.println ("Given: Precision = " + fmt.format(PR) + " and Recall " +
            fmt.format(R));
        System.out.println ("F-value = " + fmt.format(F));
    }
}
```