

# COE 212 – Engineering Programming

Welcome to Exam I  
Monday October 24, 2016

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 **105 minutes** to complete the **6** 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 helper and driver classes:

|   |  |
|---|--|
| <pre>public class Driver { public static void main(String[] args){     Helper helper = new Helper(1.66);     int dol, q, d, n, p;     dol = helper.getDol();     System.out.print("dol: " + dol);     q = helper.getQ();     System.out.print(", Q: " + q);     d = helper.getD();     System.out.print(", D: " + d);     n = helper.getN();     System.out.print(", N: " + n);     p = helper.getP();     System.out.println(", P: " + p);     String helper_str = dol+q+d+n+p+"";     System.out.print(helper_str); } }</pre> | <pre>public class Helper { private int value; public Helper(double v){     value=(int) v*100;} public int getDol(){     int output=value/100;     value%=100; return output;} public int getQ() {     int output=value/25;     value%=25; return output;} public int getD() {     int output = value/10;     value%=10; return output;} public int getN() {     int output = value/5;     value%=5; return output;} public int getP() {     return value;} }</pre> |
|---|--|

- 1) How many output lines does the execution of the **driver** class produce?
  - a. 2
  - b. 3
  - c. 4
  - d. None of the above
- 2) What output does the **first** print **statement** in the driver class produce?
  - a. dol:1
  - b. dol:2
  - c. dol:0
  - d. **None of the above**
- 3) What output does the **second** print **statement** in the driver class produce?
  - a. , Q: 1
  - b. , Q: 2
  - c. , **Q: 0**
  - d. None of the above
- 4) Which of the following can be used to store the value of the variable p from the driver class **in a String variable called str**?
  - a. **String str = (new Integer(p)) + "";**
  - b. String str = Integer.toString(p);
  - c. Both of the above
  - d. None of the above
- 5) What value does the helper\_str variable from the driver class **receive after the assignment statement** preceding the final print statement **is executed**?
  - a. "12111"
  - b. "6"
  - c. "1000"
  - d. **"1"**

- 6) Which of the following statements **correctly converts** `helper_str` **into a number**?
- `double val = Double.parseDouble(helper_str);`
  - `int val = Integer.parseInt(helper_str);`
  - Both of the above**
  - None of the above
- 7) Which of the following **correctly extracts the first character** of `helper_str`?
- `char firstChar = helper_str.charAt(1);`
  - `char firstChar = helper_str.substring(0, 1);`
  - Both of the above
  - None of the above**
- 8) Which of the following **correctly extracts the last character** of `helper_str`?
- `char lastChar = helper_str.charAt(length()-1);`
  - `Character lastChar = helper_str.charAt(length()-1);`
  - Both of the above
  - None of the above**
- 9) Which of the following replaces in `helper_str` **every occurrence** of `'1'` with `'0'`?
- `helper_str = helper_str.replace('1', '0');`**
  - `helper_str = helper_str.replace('0', '1');`
  - `helper_str = helper_str.replace(1, 0);`
  - None of the above
- 10) Assuming that `helper_str` correctly replaces every `'1'` with `'0'` in (9), which of the following can be used to **restore the original value** of `helper_str`?
- `helper = helper_str.replace('1', '0');`
  - `helper = helper_str.replace('0', '1');`
  - `helper = helper_str.replace(0, 1);`
  - None of the above**
- 11) Assuming that `helper_str` is now storing a value of `"1211"`. Which of the following correctly **removes the character '2' from** `helper_str`?
- `helper_str = helper_str.substring(0, helper_str.indexOf('2')) + helper_str.substring(helper_str.indexOf('2')+1);`
  - `helper_str = helper_str.substring(0, helper_str.indexOf('2')) + helper_str.substring(helper_str.indexOf('2'));`
  - `helper_str = helper_str.substring(helper_str.indexOf('2')) + helper_str.substring(0, helper_str.indexOf('2')-1);`
  - None of the above**
- 12) Given that `helper_str` has value `"111"` resulting from the previous question, which of the following can be used to **insert 2 at its beginning** to make it a value of `"2111"`.
- `helper_str = 2 + helper_str;`
  - `helper_str = "2" + helper_str.charAt(0) + helper_str.charAt(1) + helper_str.charAt(2);`
  - Both of the above**
  - None of the above
- 13) Given the new value of `helper_str` of `"2111"`, which of the following can be used to extract the second half of the characters from `helper_str` (i.e., `"11"`)?
- `String secondHalf = helper_str.substring(helper_str.length()/2, helper_str.length());`**
  - `String secondHalf = helper_str.charAt(helper_str.length()/2) + helper_str.charAt(helper_str.length()-1) + "";`
  - Both of the above
  - None of the above

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

1. A declared Java variable must be assigned a value before it can be used; otherwise, the compiler produces a compile-time error.

Answer: **True** False

2. The number of formal parameters in the method definition's parameter list must match the number of actual parameters in the method call; otherwise, a run-time error occurs.

Answer: True **False**

3. The following class definition results in a syntax error:

```
public class Foo {
    private int value1;
    public Foo(int value2) {value1 = value2;}
    public double getValue1() {return value1;}
    public int setValue1(int value2) {value1=value2;}
}
```

Answer: **True** False

4. The following statement outputs: NAN

```
System.out.print(Math.sqrt(-4));
```

Answer: True **False**

5. The following statement correctly stores in the variable called `val` a random floating point value between 0.0 (inclusive) and 6.0 (exclusive):

```
int val = Math.random()*6;
```

Answer: True **False**

6. The following code fragment outputs: 25.5

```
DecimalFormat fmt = new DecimalFormat("0.#");
System.out.print(fmt.format(Double.parseDouble("23.46"))+2);
```

Answer: True **False**

7. The following statement replaces in `str` every occurrence of `'i'` with `'e'`. Assume that `str` is a `String` variable that has been instantiated properly.

```
str.replace('i', 'e');
```

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. In the following statement, 2 conversion operations are performed: a conversion via promotion and a conversion via casting. Assume that `sum` and `count` are both `int` variables.

```
double avg = (double) (sum/count);
```

Answer: True **False**

10. The following code fragment outputs: 3.1

```
double pi = Math.PI();
DecimalFormat fmt = new DecimalFormat("0.#");
System.out.print(fmt.format(pi));
```

Answer: True **False**

11. The following statement correctly stores the character value `'c'` in the `String` variable called `str`.

```
String str = Character.toString('c');
```

Answer: **True** False

12. The following Java statement is syntactically valid:

```
double val = new Double(23.5);
```

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 a;     public ClassA(){a = 5;};     public void first(int b){         second(b);         a++;}     private void second(int b){         third(b-1);         a--;}     private int third(int b) {         return a/b;}     public String toString(){         return a + "";} }</pre> | <pre>public class ClassADriver {     public static void main(String[] args){          ClassA obj = new ClassA();         obj.first(1);         obj.second(2);          System.out.print("Answer is: " + obj);     } }</pre> |
|---|---|

When running the ClassADriver class, what output is produced?

- Answer is: 4
- Answer is: 5
- It doesn't compile correctly**
- It produces a run-time error
- None of the above

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

|   |   |
|---|---|
| <pre>public class ClassB {     private int a, b;     public ClassB() {         a=2; b=3;}     public void first() {         int c = a;         a = b;         b = c;}     public void second() {         int c = a+b;         a = c-a;         b=c-b;}     public String toString(){         first();         return "" + a + b;} }</pre> | <pre>public class ClassBDriver {     public static void main(String[] args){         ClassB b=new ClassB();         b.first();         b.second();         System.out.print(b);     } }</pre> |
|---|---|

When running ClassBDriver class, what output is produced?

- 23
- 32**
- It doesn't compile correctly
- It produces a run-time error
- None of the above

**Problem 4: Code analysis (15 minutes) [11 points]**

Consider the following class:

```
public class Calculus{
    private int a, b, c, d;
    public Calculus(int x, int y){
        a=x; b=a/2; c = b; d = y + a; }

    public int geta() { return a; }
    public int getb() { return b; }
    public int getc() { return c; }
    public int sum() { return a + b + c + d; }
    public String toString() {
        return "a= "+geta()+"", b= "+b+", c= " + getc() + ", d= "
+d; }}
```

This class is supplemented with the following driver class:

```
public class Driver{
    public static void main(String[] args) {
        Calculus C1 = new Calculus (8, 3);
        Calculus C2 = new Calculus (C1.sum(), C1.getc());
        int i = C1.toString().indexOf("=");
        String S = C2.toString().substring(i-1, i+2); }
}
```

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(C1);`

**a=8, b=4, c=4, d=11**

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

**1**

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

**a=**

4. `int x = C1.geta()*10 + C1.getb();`  
`int y = C2.geta()*10 + C2.getb();`  
`System.out.println("Power=" + Math.pow(x - y, 1/2));`

**Power=1.0**

5. `System.out.println(C1.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 a=2, b=3, c = 2, x = 5;`  
`x = (int)Math.pow(Math.sqrt(Math.pow(b, a)), c % x);`  
**Answer: x= 9**
- (2) `String S1 = new String("Don't waste your time with explanations");`  
`String S2 = new String("People hear only what they wanna hear");`  
`String x = S1.concat(S2).toUpperCase().substring(27, 51);`  
**Answer: x= "EXPLANATIONSPEOPLE HEAR";**
- (3) `double a = Math.pow(2, 3);`  
`double b = Math.pow(2, 4);`  
`double x = Math.pow(a + b + 1.0, 1/2);`  
**Answer: x= 1.0**
- (4) `DecimalFormat fmt = new DecimalFormat("00.##");`  
`String str = fmt.format(1.111).substring(1, 1);`  
`int x = str.length() + 1;`  
**Answer: x= 1**
- (5) `String str = "12345";`  
`boolean x = str.equals(1+'2'+ "3"+"45");`  
**Answer: x= false**
- (6) `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'**
- (7) `String str = "Engineers vs Politicians";`  
`char x = str.charAt(str.length() - str.substring(3).length());`  
**Answer: x= 'i'**
- (8) `int a=0, b=4;`  
`String str = "Good Afternoon Lebanon";`  
`String x = "\\ "+str.substring(a, b)+"\\";`  
**Answer: x= ""+Good""**
- (9) `double val1 = 12.4567;`  
`double val2 = Math.floor(12.4567*10);`  
`double x = val2 - (int) val1*10;`  
**Answer: x= 4.0**
- (10) `String x = "A photon needs 100,000 years to travel" +`  
`"from center of sun to its surface";`  
`x = x.replace('o', 'e').replace('e', 'a').replace('a',`  
`'x').substring(58);`  
**Answer: x= "tx its surfxcx"**

**Problem 6: Coding Problems (30 minutes) [39 points]**

1. Write a program called `RotatedLeft` which reads from the user a string `S1`, and then produces and prints out a new string `S2` where the first half of characters of `S1` have been moved to its end in `S2`.

**Sample run:****Enter string S1: Johnny****The resulting string S2: nnyJoh**

```
import java.util.Scanner;

public class RotatedLeft
{
    public static void main(String [] args)
    {
        String S1, S2;

        Scanner scan = new Scanner(System.in);

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

        S2 = S1.substring(S1.length()/2) + S1.substring(0,
        S1.length()/2) ;

        System.out.print("The resulting string S2: " + S2);

    }
}
```



2. Write a program called `EnclosedString` which reads from the user two strings `S1` and `S2`, and then produces a new string `S3` which consists of `S1` and `S2` combined such that `S2` is put in the middle of `S1`.

**Sample run:**

**Enter string S1: <<>>**

**Enter string S2: John**

**The resulting string S3: <<John>>**

```
import java.util.Scanner;

public class EnclosedString
{
    public static void main(String [] args)
    {
        String S1, S2, S3;

        Scanner scan = new Scanner(System.in);

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

        System.out.print("Enter string S1: ");
        S2 = scan.nextLine();

        S3 = S1.substring(0, S1.length()/2) + S2 +
            S1.substring(S1.length()/2);

        System.out.print("The resulting string S3: " + S3);

    }
}
```

3. Write a program called `SimpleAdverb` which reads from the user a word, and then prints `true` if the word represents an adverb and `false` otherwise. To simplify the problem, we will consider a word to be an adverb if it ends with `"ly"` (such as *oddly*, *dangerously*, *happily*, and *heavily*, etc.). NOTE: you are not allowed to use conditional operators (such as *if-else*).

**Sample run:**

**Enter a word: Obviously**  
**The word is an adverb: true**

```
import java.util.Scanner;

public class SimpleAdverb
{
    public static void main(String [] args)
    {
        final String ending = "ly";

        Scanner scan = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = scan.nextLine();

        boolean adverb = word.substring(word.length() -
2).equals(ending);

        System.out.print("The word is an adverb: " + adverb);

    }
}
```