

COE 211/COE 212 – Computer Programming/Engineering Programming

Welcome to
The Comprehensive Final Exam
Monday June 04 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 **150 minutes** to complete the **eight** 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: Inheritance (10 minutes) [10 points]

For each question choose the **single** right answer.

- 1) Which of the following statements is **true** about constructors?
 - a. Constructors must have a return type
 - b. Constructors are not inherited**
 - c. Constructors of parent class cannot be invoked by the child class
 - d. None of the above
- 2) All classes in Java are directly or indirectly subclasses of which of the following classes
 - a. Wrapper
 - b. String
 - c. Class
 - d. None of the above**
- 3) Aside from permitting inheritance, the “protected” visibility modifier:
 - a. permits access to the protected item by any class defined in the same package**
 - b. permits access to the protected item by any other class
 - c. permits access to the protected item by any parent class
 - d. None of the above
- 4) Which of the following statements establishes PayType as a parent class for Salaried?
 - a. public class Salaried implements PayType
 - b. public class Salaried derivedFrom(PayType)
 - c. public class PayType extends Salaried
 - d. None of the above**
- 5) Method overriding
 - a. Occurs when a class defines 2 methods with same name but with different signatures
 - b. Occurs when a parent class is not sure about a method definition so it leaves the definition for the child class
 - c. Occurs when a child class defines a method with the same name and signature as a method in the parent class**
 - d. None of the above
- 6) If a class called B is used to derive a new class called A, then
 - a. Public and private members of B can be used in A
 - b. Public members in B can be used in A, but protected members cannot be accessed in A
 - c. Neither public nor private members in B can be directly referenced in A
 - d. None of the above**
- 7) Which of the following methods is supported by every class created in Java?
 - a. compareTo
 - b. charAt
 - c. equals**
 - d. None of the above
- 8) Consider the following statement: `super () ;` This statement
 - a. calls the method super defined in the current class
 - b. calls the method super as defined in java.lang
 - c. calls the constructor as defined in the current class
 - d. calls the constructor as defined in the parent of the current class**
- 9) When designing a class hierarchy, it is important that common features be kept
 - a. higher in the class hierarchy**
 - b. lower in the class hierarchy
 - c. near the middle of the class hierarchy
 - d. in the Object class
- 10) The original class that is used to derive a new class via inheritance is called:
 - a. A superclass
 - b. A base class
 - c. A subclass
 - d. Both (a) and (b)**

Problem 2: Arrays (15 minutes) [11 points]

For each question choose the **single** right answer.

- 1) Which of the following statements prints the **third** command line argument out?
- System.out.println(args[3]);
 - System.out.println(args[2]);**
 - System.out.println(args[1]);
 - None of the above

In what follows, assume that **states** is a two dimensional array of **Strings** that is filled up to capacity with the following values:

states →

AL	CA	HI
FL	MA	MI
MN	TX	NY
NC	ND	OH

- 2) What is returned by `states[3][2]`?
- "TX"
 - "MI"
 - "OH"**
 - None of the above
- 3) What is the value of `states.length`?
- 3
 - 12
 - 4**
 - None of the above
- 4) The statement `System.out.println(states[4][2]);` will
- output ND
 - output OH
 - cause an ArrayOutOfBoundsException to be thrown**
 - None of the above
- 5) What is returned by `states[states[0].length-2][states[1].length-2].charAt(0)`?
- 'F'
 - 'A'
 - 'M'**
 - None of the above
- 6) Which of the following statements can be used to declare `states`?
- String[][] states = new String[3][2];
 - String[][] states = new String[4][3];**
 - String[][] states = new String[2][3];
 - String[][] states = new String[3][4];
- 7) What is returned by `states[1][0].length()`?
- 2**
 - 4
 - 3
 - None of the above

8) What is the output of the following code fragments?

```
int[] zip = new int[4];
zip[0] = 7;
zip[1] = 1;
zip[2] = 4;
int j = zip[1];
System.out.println(zip[j+1]+zip[3]);
```

- a. 5
- b. 7
- c. 4**
- d. None of the above

```
int[] data = {0, 3, 7, 9, 12, 14};
System.out.println(data[data[1]]);
```

- a. 7
- b. 9**
- c. 12
- d. 14

```
int[] arr = new int[5];
int j = 2;
arr[0] = 4;
arr[1] = 5;
arr[2] = 6;
arr[3] = 7;
int j = arr[j];
System.out.println(arr[(int)Math.floor((j+1)/2.0)]);
```

- a. 5
- b. 7**
- c. 4
- d. None of the above

```
ArrayList players = new ArrayList();
players.add("Messi"); players.add("Ronaldo");
int location = players.indexOf("Messi");
players.add(location, "Rooney");
players.add("Zidane");
players.remove(1);
System.out.println(players.get(2));
```

- a. Messi
- b. Ronaldo
- c. Zidane**
- d. Rooney

Problem 3: Code analysis (25 minutes) [14 points]

- 1) Fill in the values for the array called world after the corresponding Java code has run. You must enter your final answers into the boxes provided.

```
int[] world={19, 23, 18, 6, 3};
int low, hi=world.length - 1, temp;
for (int i=0; i < hi; i++) {
    low = i;
    for(int j=i+1; j < (hi+1); j++) {
        if(world[j] < world[low])
            low = j;
    }
    temp = world[low];
    world[low] = world[i];
    world[i] = temp;
}
```

Answer: world →

23	19	18	6	3
----	----	----	---	---

```
int[][] A = {{2, 5}, {3, 6}, {4, 7}};
int[][] world = new int[A[1].length][A.length];
for(int i=0; i<A.length; i++)
    for(int j=0; j<A[0].length; j++)
        world[j][i] = A[i][j];
```

Answer: world →

- 2) 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()?

<pre>public class Confuse{ private int x; private int y; public Confuse(){ x=3; y=5; } private void first(int a){ x = a; x++; } private void second(int s,int t){ setXY((y+s), (x-t)); } }</pre>	<pre>private void setXY(int s,int t){ x = s; y = t; } private void display(){ System.out.println(x+y); } public void startUp() { first(x+1); second(y,x); display(); }</pre>
--	--

- 1
- 3
- 10**
- None of the above

Problem 4: True or false (10 minutes) [10 points]

1. Two methods in Java can have the same name and the same number of parameters.
Answer: **True** False

2. A non-static method can access static instance variables.
Answer: **True** False

3. The `length()` method is used for both strings and arrays. You must be sure to include the parenthesis in both cases.
Answer: True **False**

4. To swap the 3rd and 4th elements in the array of `int` values called `list`, you would do:

```
int temp = list[2];
list[2] = list[3];
list[3] = temp;
```

 Answer: **True** False

5. If a program attempts to access an element outside of the range of the array indexes, a run time error occurs.
Answer: **True** False

6. Given the following method definition:

```
int foo(int... list) {
    int sum=0, n = list.length;
    for(int i=0; i < n; i++)
        sum+=list[i];
    return (n>0) ? sum/n : 0;}

```

 After the following statement executes: `double x = foo(2,5);` `x` will hold a value of 3.5.
 Answer: True **False**

7. The following code stores in the variable called `sum` the sum of the positive odd integers that are less than 10:

```
int sum = 0;
for(int x = -1; x <= 10; x+=2)
    if(x > 0) sum += x;
```

 Answer: **True** False

8. If a class called `Transaction` contains a static variable called `howMany`, then each instance of the class will get its own copy of the variable `howMany`.
 Answer: True **False**

9. A child class is not allowed to define a method that has the same name and parameter list as a method in the parent class.
 Answer: True **False**

10. A derived class has access to all of the public and protected members of the parent class.
 Answer: **True** False

Problem 5: Method definition (15 minutes) [8 points]

You are given below the headers of 3 methods called `countUpper`, `arrayCountPrime`, and `reverse`. Your job is to complete the definition of each one of these methods as per the provided guidelines.

1. **`countUpper`** is a method that accepts a `String` parameter and returns the number of upper case letters found in that `String`.

```
public int countUpper(String phrase) {
```

```
    int cnt=0;
    for (int i=0;i<phrase.length();i++)
    {
        char c=phrase.charAt(i);
        if (c>='A'&&c<='Z')
            cnt++;
    }
    return cnt;
}
```

2. **`arrayCountPrime`** is a method that returns the number of prime values found in the array of `int` parameter called `arr`.

```
public int arrayCountPrime(int[] arr) {
```

```
    int cnt=0;
    for (int i=0;i<arr.length;i++)
    {
        boolean isPrime=true;
        int n=arr[i];
        for (int j=2;j<=i/2;j++)
        {
            if (n%j==0)
            {
                isPrime=false;
                break;
            }
        }
        if (isPrime)
            cnt++;
    }
    return cnt;
}
```

3. **`reverse`** is a method that accepts as a parameter a `String` called `str` and returns a new `String` that contains the characters of `str` in reverse order.

```
public String reverse(String str) {
```

```
    String s="";
    for (int i=str.length()-1;i>=0;i--)
        s+=str.charAt(i);
    return s;
}
```

Problem 6: Completing code fragments (25 minutes) [10 points]

(1) Complete the definition of the class Student given below. Each student has a **name**, an array called `scores` that can hold up to **ten integer quiz scores** and a variable called `count` indicating the number of scores added so far to the `scores` array (`count` is initially 0).

```
public class Student {
    // Put the instance variables here
    String name;
    int[] scores;
    int cnt;

    // Constructor initializing all instance variables
    public Student(String theName) {
name=theName;
scores=new int[10];
cnt=0;
    }
    // the method addScore inserts the specified score
    // into the scores array at the specified index.
    public void addScore(int score, int index) {

scores[index]=score;

    }
    // the method getMinimum returns the minimum quiz
    // grade currently stored for the student in scores.
    public int getMinimum() {
int min=scores[0];
for(int i=1;i<cnt;i++)
    if (scores[i]<min)
        min=scores[i];
return min;

    }
    // Returns a String with name and minimum quiz grade
    public String toString() {
return name+" "+getMinimum();

    }
} // end Student
```

Problem 6 continued:

(2) Below is given a class named Person. A person has a name and a year of birth.

```
public class Person {
    protected String name;
    private int year;
    public Person(String n, int y) {
        name = n; year = y;
    }
    public String getYear() {return year;}
    public String toString() {
        return name + " " + year;
    }
}
```

Complete the partial definition given below for a subclass called Coach derived from the Person class. Note that a coach object also has a ranking for his/her team (an int value). Complete the constructor, equals, getter and toString methods of the Coach class as noted by the comments highlighted in bold.

```
public class Coach extends Person{
    // instance variable representing ranking goes here.
    int rank;

    // constructor must invoke the constructor of parent
    // class.
    public Coach(String theName,int birthYear,int rank) {
        super(theName,birthYear);
        this.rank=rank;
    }
    // A method that returns the ranking.
    public int getRank() {
        return rank;
    }
    // two Coaches are equal if they have the same
    // ranking, the same name, and the same year of birth.
    public boolean equals(Object other) {
        Coach tmp=(Coach)other;
        return(tmp.name.equals(name)&&(year==tmp.getYear())&&(
tmp.rank==rank));
    }
    public String toString ( ) {
        //Invoke the Person's version of toString and then
        //concatenate the ranking to the returned String.
        return super.toString()+rank;
    }
}
```

Problem 7: Writing code segments (15 minutes) [7 points]

Write each of the Java code segments requested below.

Example: write a **one-line statement** declaring a string storing your name.

Answer: `String name = "wissam";`

- a) Write a **one-line statement** that defines the header of a method called `average` accepting a **one-dimensional** array of integers.

```
Public double average(int[] a)
```

- b) Write a **one-line statement** declaring, creating, and initializing the following **two-dimensional** array of temperatures: $temps = \begin{bmatrix} 64.5 & 53.2 \\ 23.7 & 45.8 \end{bmatrix}$

```
Double[][]t ={{ 64.5,53.2},{23.7,45.8}};
```

- c) Write **one or more Java statements** to print each of the values stored in the array `temps` (from part (b)) on a separate line.

```
For (int i=0;i<2;i++)
  For(int j=0;j<2;j++)
    System.out.println(temps[i][j]);
```

- d) Write **one or more Java statements** to print the smallest number found in the array `temps` (from part (b)).

```
Int min=temps[0][0];
For (int i=0;i<2;i++)
  For(int j=0;j<2;j++)
    If (temps[i][j]<min)
      min=temps[i][j];
System.out.println(min);
```

- e) Write a **code segment** to define a method called `avgMatrix` that receives a **two-dimensional** array of `int` values and returns their average.

```
Public avgMatrix(int[][] a)
Int sum=0;
Int cnt=a.length*a[0].length;
For(int i=0;i<a.length;i++)
  For(int j=0;j<a[i].length;j++)
    sum+=a[i][j];
return (double)sum/cnt;
```

Problem 8: Coding (35 minutes) [30 points]

1. Write a Java program that reads ten integers from the user. Your code should compute their average and should output how many of the input numbers are strictly greater than the computed average.

```
import java.util.Scanner;

public class RandomNumbers {

    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        int[] nums=new int[10];
        for (int i=0;i<10;i++)
            nums[i]=scan.nextInt();
        double av=getAverage(nums);
        int cnt=0;
        for (int i=0;i<10;i++)
            if (nums[i]>av)
                cnt++;
        System.out.println(cnt);
    }
    public static double getAverage(int[] a)
    {
        int sum=0;
        for (int i=0;i<a.length;i++)
            sum+=a[i];
        return (double)sum/a.length;
    }
}
```

Write a Java program that loads 200 **random integers** between 0 and 99 and then sorts the elements of the array in an **ascending order** before displaying the sorted version of the array on the screen.

```
public class RandomInts {  
  
    public static void main(String[] args) {  
        int nums[]=new int[200];  
        for (int i=0;i<nums.length;i++)  
            nums[i]=(int)(Math.random()*100);  
        sort(nums);  
        for (int i=0;i<nums.length;i++)  
            System.out.println(nums[i]);  
    }  
    public static void sort(int []a)  
    {  
        for (int i=0;i<a.length;i++)  
        {  
            int min=i;  
            for (int j=i+1;j<a.length;j++)  
                if (a[j]<a[min])  
                    min=j;  
            int tmp=a[i];  
            a[i]=a[min];  
            a[min]=tmp;  
        }  
    }  
}
```

1. Write a Java program called **binaryToDecimal** that reads from the user a string consisting of a sequence of '0' and '1' characters. The obtained string is a binary representation of a decimal (base-10) integer number. Display on the screen the decimal integer value corresponding to that input string. Recall that the binary number system is also called the base-2 number system and has only two digits to represent values, namely '0' and '1'. So, for example, 1100 which is a binary number is equivalent to 12 in decimal since:
 $(1100)_2 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = (12)_{10}$

Sample output:

Enter a binary value: 110

The integer corresponding to "110" is: 6

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

```
public class BCD {

    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.print("Enter a binary number: ");
        String bin=scan.next();
        double dec=0;
        for (int i=0;i<bin.length();i++)
        {
            if (bin.charAt(i)=='1')
                dec+=Math.pow(2, bin.length()-1-i);
        }
        System.out.println("The decimal value is: " +dec);
    }
}
```

2. Consider a three-dimensional array of the form $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

Write a Java program which finds the transpose of A denoted by A^T . Note that A^T can be derived from A by interchanging the rows and columns of A. That is, the rows of A become the columns of A^T and the columns of A become the rows of A^T .

$$A^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

Note that the program should read the values of a, b, c, d, e, f, g, h, and i from the user and then display the entries of A^T in a table format with 3 rows and 3 columns.

```
import java.util.Scanner;

public class Transpose {

    public static void main(String[] args) {
        int[][]a=new int[3][3];
        Scanner scan=new Scanner (System.in);
        for (int i=0;i<3;i++)
            for (int j=0;j<3;j++)
                a[i][j]=scan.nextInt();
        for (int i=0;i<3;i++)
            for (int j=i+1;j<3;j++)
                {
                    int tmp=a[i][j];
                    a[i][j]=a[j][i];
                    a[j][i]=tmp;
                }
        for (int i=0;i<3;i++)
        {
            for (int j=0;j<3;j++)
                System.out.print(a[i][j]+" ");
            System.out.println();
        }
    }
}
```