COE 212 – Engineering Programming

Welcome to the Final Exam
Monday May 18, 2015

Instructors: Dr. Joe Tekli
Dr. George Sakr
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 **minutes** to complete 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: Arrays (**15 minutes)** [12 points]

1) Which of the following statements about arrays are **false**?
   A. An array is a dynamic collection of values that have the same type
   B. An array element is accessed by specifying its index
   C. The length of an array called `arr` is determined by the expression `arr.length();`
   D. The second element of array `arr` is specified by `arr(1)`
   a. A, B, D
   b. C, D
   c. **A, C, D**
   d. B, D

2) Consider the following:
```
int[] arr = new int[4];
arr[2] = 1; arr[1] = 2; arr[3] = 3;
```
The value of `arr[arr[3]-arr[1]-arr[2]]` is
   a. **`0`**
   b. `3`
   c. `2`
   d. None of the above

3) Which of the following declares and populates the array called `arr` with the first **5 integer** values?
   a. `int arr[5] = {1, 2, 3, 4, 5};`
   b. `int arr[] = {1; 2; 3; 4; 5};`
   c. `arr = {1, 2, 3, 4, 5};`
   d. **None of the above**

4) Attempting to access an array element outside of the bounds of an array causes a
   a. Syntax error called `ArrayOutOfBoundsException`
   b. Syntax error called `ArrayElementOutOfBoundsException`
   c. Syntax error called `ArrayException`
   d. **None of the above**

5) Consider an array called `items`, which contains the values `0`, `2`, `4`, `6`, and `8`. If the method `foo` given below is called as follows: `foo(items, items[2]);` what values are stored in `items` after the method finishes executing?

```
public void foo(int[] passedArray, int value) {
   passedArray[value] = 12;
   value = 5;
}
```

   a. `0, 2, 5, 6, 12`
   b. `0, 2, 12, 6, 8`
   c. `0, 2, 4, 6, 5`
   d. **None of the above**

6) Which statement below initializes an array called `items` to contain 3 rows and 2 columns?
   a. `int[] items = {{2,4}, {6,8}, {10,12}};`
   b. `int[] items = {{2, 6, 8}, {4, 8, 12}};`
   c. `int[] items = {2,4}, {6,8}, {10,12}`
   d. **None of the above**

7) Which set of statements below sums up the items in each row of a two dimensional array called `items` and displays each row's `total`?

   a.
```
for (int row = 0; row < items.length; row++){
      int total = 0;
      for (int column = 0; column < items[row].length; column++)
         total += items[row][column];
      System.out.println("Total: " + total);}
```

b.
```
int total = 0;
   for (int row = 0; row < items.length; row++){
      for (int column = 0; column < items[row].length; column++)
         total += items[row][column];
      System.out.println(“Total: " + total);}
```

c.
```
int total = 0;
   for (int row = 0; row < items.length; row++){
      for (int column = 0; column < items[column].length; column++)
         total += items[row][column];
      System.out.println("Total: " + total);}
```

d.
```
for (int row = 0; row < items.length; row++){
      int total = 0;
      for (int column = 0; column < items[column].length; column++)
         total += items[row][column];
      System.out.println("Total: " + total);}
```

8) Which of the following returns the number of elements stored in an `ArrayList` called `list`?
   a. `list.length`
   b. `list.size`
   c. `list.length()`
   d. **None of the above**

9) Which of the following correctly declares an `ArrayList` of integer values called `list`?
   a. `ArrayList<int> list`
   b. `ArrayList<Int> list`
   c. Both of the above
   d. **None of the above**

10) What does the following code do? Assume that `items` is an array of `int` values and that `temp` is an `int` variable previously initialized to `0`.
```
for(int j=0; j < items.length; j++)
      if(items[j] < temp) temp = items[j];
```
   a. It finds the largest value in the array and stores it in `temp`
   b. It finds the smallest value in the array and stores it in `temp`
   c. It finds the last value in the array that is greater than `temp`
   d. **None of the above**

11) Consider the following statements:
```
int[] a = {3, 6, 8, 9};
int[] b = a;
```
Which of the following is true?
   a. The array called `b` contains a copy of the elements of the array called `a`
   b. **The variables `a` and `b` are aliases of each other**
   c. Both of the above
   d. None of the above

12) Given the following declarations, which of the following variables are not arrays?
```
int[] a, b;
int d[], c, e;
```
   a. `b, c, e`
   b. `a, b, c, e`
   c. `a, c, e`
   d. **None of the above**

Problem 2: True or false questions (**15 minutes)** [12 points]

1. The methods and variables declared as private in the parent class are not inherited by the child class and as such cannot be referenced by name inside the child class.

Answer: True **False**

2. The constructor of a parent class can be referenced from within the subclass via the "`this`" reserved word.

Answer: True **False**

3. It is possible for a Java class to implement an interface without providing a definition for one of the abstract methods present in that interface.

Answer: True **False**

4. In Java, a class may have multiple children but can have only one parent class since Java supports multiple inheritance.

Answer: True **False**

5. The following code fragment prints out a value of `4`

```
int[] x = new int[5];
int i;
for(i=0; i < x.length; i++)
      x[i] = i;
System.out.println(x[i]);
```

Answer: True **False**

6. The code fragment below prints the following out: `1 1 2 3 4 5`

```
int[] list = {1, 2, 3, 4, 5, 6};
for(int i=0; i < list.length; i++)
      list[i+1] = list[i];
for(int i=0; i < list.length; i++)
      System.out.print(list[i] + " ");
```

Answer: True **False**

7. Consider the following statement: `boolean[][] arr=new boolean[5][7]`. The values of `arr.length` and `arr[7].length` are `5` and `7` respectively.

Answer: True **False**

8. The following code fragment constructs a new `String` called `str` that contains all the characters stored in the array called `arr`.

```
char[] arr = {'h', 'e', 'l', 'l', 'o'};
String str;
for(int i=0; i < 5; i++){
      str+=arr[i];}
```

Answer: True **False**

9. The following method sorts the elements of the array called `arr` in an ascending order.

```
public static void foo(int[] arr) {
      int j, k;
      for(int i=1; i<arr.length; i++) {
            k = arr[i];
            j=i;
            while(j>0 &&arr[j]<arr[j-1]){
                  arr[j]=arr[j-1];
                  j--;}
            arr[j] = k;}}
```

Answer: **True** False

10. The following code fragment prints out all the elements of the `ArrayList` called `list`.

```
for(int j=0; j<list.size(); j++)
      System.out.print(list.get(j)+ " ");
```

Answer: **True** False

11. The following method prints out `Joe` if `A={"Wissam", "Joe", "Georges"}`

```
public static void foo(String[] A) {
      ArrayList<String> list = new ArrayList<String>();
      for(int i=0; i<A.length; i++)
            list.add(A[i]);
      while(list.size() > 1) {
            for(int i=1; i<=2; i++)
                  list.add(list.remove(0));
            list.remove(0);
      }
      System.out.print(list.get(0));}
```

Answer: **True** False

12. The following binary search implementation works properly only if the input array `arr` is sorted in ascending order.

```
public static boolean foo(int[] arr, int target) {
      int min = 0, max = arr.length-1, mid;
      boolean found=false;
      while(!found && min <= max) {
            mid = (min+max)/2;
            if(arr[mid]== target) found = true;
            else if(arr[mid]>target) max = mid-1;
            else min = mid+1;}
      return found;}
```

Answer: **True** False

Problem 3: Multiple Choice Questions (**15 minutes**) [10 points]

1) Consider the method given below. Its output can be best described as:

```
void method1(int number){
      int count = 0, denom, value = number;
      while(value > 0) {
            count++;
            value = value/10;}
      denom = (int) Math.pow(10, count);
      while(number>0) {
            System.out.print(""+(number/denom));
            number%=denom;
      }
}
```

a. The input digits in reverse order
b. A tenth of the original input
c. All of the input digits except the rightmost digit in the original order
d. The input digits in the original order
e. None of the above

2) What is the output when `methodF` is called as follows:
`methodF("AManAPPanama");`

```
private String mySubstring(String words,
      int i, int j){
      if (i<=j)
            return words.substring(i,j);
      else
            return words.substring(j,i);}

public void methodF(String words){
      int x = words.length();
      int i=0;
for (int j=x-1; i<x/2; i++){
      words = swapLetter(words, i, j);
      j--;
}
System.out.println( words);}
```

```
public String swapLetter(String words,
      int i, int j){
      int x = words.length();
      String lPiece,mPiece,rPiece;
      String lChar, rChar;
      lPiece = mySubstring(words, 0,i);
      lChar = mySubstring(words, i,i+1);
      mPiece = mySubstring(words, i+1,j);
      rChar =mySubstring(words, j, j+1);
      rPiece = mySubstring(words, j+1,x);
      words = lPiece + rChar +
            mPiece + lChar + rPiece;
      return words;
}
```

Its output is:

a. The content of the input String in reverse order
b. The content of the input String in the original order
c. The content of the original String with the characters shifted to the right by one position
d. The content of the original String with the characters shifted to the left by one position
e. None of the above

Problem 4: Evaluating Java Expressions (**25 minutes**) [16 points]

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

(1)
```
int [] A = {2, 4, 6, 8, 10}; int x=0;
int i = A.length;
while(i > 1){
   if(i%3 == 0)  {x = A[i-1];}
   else { x = A[i-1]++;}
   i--;
}
```
Answer: x= **4**

(2)
```
int m = 2; int x=0;
int [] T = {1, 1, 1};
for ( int n = 0; n < T.length; n++)
{ for (int i=1; i<=m; i++)
T[n] *= i; m++; x+= T[n];}
```
Answer: x= **32**

(3)
```
int [] A = new int[4];
for (int i = 0; i < A.length; i*=2) {
A[i] *= i; i++;} int x=0;
for (int i = 0; i < A.length/2; i++) {
x += A[i]; A[i]++;}
```
Answer: x=**0**

(4)
```
ArrayList<Integer> T = new ArrayList<Integer>();
int x = 3;
for (int i = 0; i<= 2; i++ )
T.add(i * 2);
x += T.get(T.size()-1);
```
Answer: x= **7**

(5)
```
String x = "Planet of the Apes", x1 = "";
for(int i=0; i< x.length(); i++)
if(x1.indexOf(x.charAt(i)) == -1)
x1 += x.charAt(i);x = x1;
```
Answer: x= **“Planet ofhAps”**

(6)
```
String S = "Patriotism is the virtue of the vicious";
Scanner scan  = new Scanner (S); int x=0;
String S1 = "aoieuAOIEU";
while (scan.hasNext()){
String S2 = scan.next();
for(int i=0; i< S2.length(); i++)
if (S1.indexOf(S2.charAt(i)) != -1) x++;}
```

Answer: x= **15**

(7)
```
int[] A= {5, 0, 7}; int[] B = {2, 3};
ArrayList<Integer> T = new ArrayList<Integer>();
String x = "";
for(int i=0; i< A.length; i++)
    for(int j=0; j< B.length; j++)
       if(A[i]%B[j]==0)  T.add(A[i]);
          for(int i=0; i< T.size(); i++)
             x += T.get(i); x +=7;
```
Answer: x= **"007"**

For each of the following code fragments, what are the values of the elements of array T after the statements are executed? (9 pts)

(8)
```
int[] R = {1, 2, 3}; int[] S = {5, 6, 7};
int[] T = new int[R.length*2];
int n=0, m=0;
for(int i=0; i< S.length*2; i++)
if(i%2 == 0)  {T[i] = R[n]; n++;}
else {T[i] = S[m]; m++;}
```

T

Indexes	0	1	2	3	4	5
Values	**1**	**5**	**2**	**6**	**3**	**7**

(9)
```
int [][] T = {{1, 2, 3}, {4, 5, 6}}; int x;
for(int j=0; j< T.length; j++)
{
    x = T[0][j]; T[0][j] = T[1][j];
    T[1][j] = x;
}
```

T

Indexes	0	1	2
0	**4**	**5**	**3**
1	**1**	**2**	**6**

(10)
```
int[][] T = {{1, 2, 3}, {4, 5, 6}};
for (int i = 0; i < T.length; i++)
  for(int j=0; j<T.length; j++)
    T[j][i]++;
```

T

Indexes	0	1	2
0	**2**	**3**	**3**
1	**5**	**6**	**6**

Problem 6: Coding (**50 minutes**) [50 points]

1. Write a program called `MinDistance`, which reads from the user the elements of an array of integers whose size is set by the user with a minimum allowable size of 2. The program then prints out the minimum difference between any two consecutive numbers in the array (see sample output below for more information).

   Sample output;
   Enter size of array (minimum 2): 3
   Enter element #0: 5
   Enter element #1: 7
   Enter element #2: 15
   Minimum distance: 2

```
import java.util.Scanner;
class  MinDistance{

    public static void main(String [] args){

    Scanner scan = new Scanner(System.in);

    int N;

    do{
          System.out.println("Enter size of array” +
                             “ (minumum 2 elements)");
          N = scan.nextInt();
      } while (N <2);

    int[] T = new int[N];

    for (int i=0; i<T.length; i++)
    {
        System.out.print("Enter element #" + i);
        T[i] = scan.nextInt();
    }
    int min = Math.abs(T[0] - T[1]);

    for (int i = 1; i < T.length - 1; i++)
        if ( Math.abs( T[i] - T[i+ 1]) < min)
               min = Math.abs(T[i] - T[i+ 1]);

    System.out.print(“Minimum distance:” + min);
  }
}
```

2. Write a program called `Histogram` that keeps on reading integer numbers between 0 (inclusive) and 10 (inclusive) until the end user enters a negative number (sentinel value). Your program should count the number of occurrences of each input value and then print out those input numbers occurring 1 or more times along with their number of occurrences.

   Sample output;
   Enter the integers between 0 and 10: 2 5 6 5 4 3 2 0 -1
   0 occurs 1 time
   2 occurs 2 times
   3 occurs 1 time
   4 occurs 1 time
   5 occurs 2 times
   6 occurs 1 time

```
import java.util.Scanner;
class  Histogram{

public static void main(String [] args){

 Scanner scan = new Scanner(System.in);

 int [] T = new int[11];

 System.out.println("Enter the integers between 0 and 10:");
 int x = scan.nextInt();

 while (x>0) {

      if(x >10) {
            System.out.println("Wrong input! Integer should be " +
                                    "comprised between 0 and 10! Try again:");
            x = scan.nextInt();
      }
      else
      {
            T[x]++;
            x = scan.nextInt();
      }
   }

for (int i = 1; i < T.length; i++)
      System.out.println(i + " occurs " + T[i] +
                         ((T[i] == 1) ? " time" : " times"));
  }
}
```

3. Write a program that reads from the user a sequence of integer values whose number is specified by the end user. Your program should then print an output message indicating whether or not the input sequence contains four back to back occurrences of the same value. Note that if multiple numbers within the sequence occur 4 times in a row, make sure to print only the first one out (In the sample output below, 5 and 6 occurred 4 times in a row and therefore, only 5 was printed out).

   Sample output;
   Enter nb of values: 11
   Enter the values: 3 4 5 5 5 5 4 6 6 6 6
   The sequence contains four back to back occurrences of 5

```
import java.util.Scanner;

class  Back2Back{

public static void main(String [] args){

Scanner scan = new Scanner(System.in);

int N;
do{
   System.out.print("Enter nb of values: ");
   N = scan.nextInt();
while(N<=0);

int [] T = new int[N];

System.out.println("Enter the values:");

for(int i=0; i<N; i++)
    T[i] = scan.nextInt();

boolean b2b = false;
int j;

for (j=0; j<= N-4 && !b2b ; j++)

{
     if (T[j] == T[j+1] && T[j] == T[j+2] && T[j] == T[j+3])
     // Could have used an inner loop here
     b2b = true;
}

if(b2b) System.out.println("The sequence contains four " +
                           "back to back occurrences of:" + T[j]);
else System.out.println("The sequence does not contain four " +
                        " back to back occurrences");

  }
}
```

4. Write a program that reads from the user a sequence of integer values whose number is specified by the end user. Your program should then print an output message indicating whether or not the input sequence is sorted in an ascending order.

   Sample output;
   Enter nb of values: 8
   Enter the values: 10 1 5 16 61 9 11 1
   The input list of values is not sorted

```
import java.util.Scanner;

class  Sorted{

public static void main(String [] args){

  Scanner scan = new Scanner(System.in);

  int N;
  do{
     System.out.print("Enter nb of values: ");
     N = scan.nextInt();
  while(N<=0);

  int [] T = new int[N];

  System.out.println("Enter the values:");
  T[0] = scan.nextInt();

  boolean sorted = true;

  for(int i=1; i<N && sorted; i++)
  {
      T[i] = scan.nextInt();
        if (T[i-1] > T[i]) sorted = false;
  }

  if(sorted) System.out.println("The input list of values is sorted");
  else System.out.println("The input list of values is not sorted");

  }
}
```

5. Write a program called `Anagram` which reads from the user two strings `S1` and `S2`, and then determines whether or not they are anagrams of each other. Two strings `S1` and `S2` are said to be anagrams of each other if the rearrangement of the characters of `S1` produces `S2` and vice versa. To test whether two Strings S1 and S2 are anagrams of each other, use the following procedure:

    a. Convert `S1` and `S2` to lowercase.
    b. Create a first array called `arr1` that would hold the number of times each alphabetic character appears in `S1`.
    c. Create a second array called `arr2` that would hold the number of times each alphabetic character appears in `S2`.
    d. Then compare the content of `arr1` to the content of `arr2`. If they are identical then `S1` and `S2` are anagrams. Note that two arrays are said to be identical if they contain the same set of values in the same order.

Sample output;
Enter the first string: Debit card
Enter the second string: Bad credit
The strings are Anagrams

```
import java.util.Scanner;
class  Anagrams{

public static void main(String [] args){

Scanner scan = new Scanner(System.in);

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

System.out.print("Enter the second string: ");
String S2 = scan.nextLine();

S1 = S1.toLowerCase();
S2 = S2.toLowerCase();

final int N = 26;   // # of lower case characters in English alphabet

int [] arr1 = new int[N];
int [] arr2 = new int[N];

for(int i=0; i<S1.length(); i++)
{
    char c = S1.charAt(i);

    if(c != ' ')
    {
        int x = S1.charAt(i) - 'a';
        arr1[x]++;
    }
}
```

```
for(int j=0; j<S2.length(); j++)
{
   char c = S1.charAt(j);
   if (c != ' ')
   {
   int y = S1.charAt(j) - 'a';
   arr2[y]++;
   }
}

boolean anagrams = true;

int p=0;
while(anagrams && p < N)
{
    if(arr1[p] != arr2[p]) anagrams = false;
    else p++;
}

if(anagrams) System.out.println("The strings are anagrams");
else System.out.println("The strings are not anagrams");


  }
}
```