

Lehigh Username (use upper case letters): @lehigh.edu

Put your name on each page, or risk having an incompletely graded exam.
Raise your hand to notify a proctor if you have a question. **No talking, no tapping or other noise. No cell phones, calculators or other electronics.**

Read the entire exam and **answer the questions that seem easiest first.** Write answers in the box when provided. **There are 250 points on this exam.**

Grader Use Only	
Question 1:	/30
Question 2:	/30
Question 3:	/20
Question 4:	/20
Question 5:	/30
Question 6:	/20
Question 7:	/20
Question 8:	/30
Question 9:	/30
Question 10:	/20
Exam Total:	

Lehigh Username (use upper case letters): @lehigh.edu

total score

1) To the left below are a sequence of statements that are executed in order. State the value of the variable `out` after each statement of the form "`out+=. . .`" is executed, in the box indicated by the arrow. Changes to `out` are cumulative. (3 points each, 30 points total)

```
String out="",side="abc",another="abcbar";  
side+="bar";  
out+=(2+3*4);
```

```
out+="\n:"+2+3*4;
```

```
out+="one + two"+3;
```

```
out+=" "+(1+3)+(2==3 || 2*3==6 && 4<3);
```

```
out+=(2+3*4/5%3);
```

```
out+="\n:"+2+3.0*4/5%3;
```

```
out+='!';
```

```
out+=side==another;
```

```
out+=":"+side.equals(another);
```

```
out+="\nHave a good summer";
```

Lehigh Username (use upper case letters): @lehigh.edu

total score

2) I showed the code below to my friend. It compiles without error. She pointed out that the code had the potential for five run-time errors, **when given certain input**. Using the line numbers that (added for your reference), **state the line number, explain the error, and state what the input would have to be to cause each error**. There are **five** runtime errors. (6 points each, 30 points total)

```

1  public static void q1() {
2      Scanner scan = new Scanner(System.in);
3      double longitude[], latitude[] = new double[10];
4      int n;
5      double sum = 0;
6      System.out.print("Number of pairs to enter?");
7      n = scan.nextInt();
8      longitude = new double[n];
9      for(int j = 0; j < n; j++){
10         longitude[j] = getDub(scan);
11         latitude[j] = getDub(scan); }
12         do{
13             n--;
14             System.out.println( longitude[n] + "," +
15                               latitude[n]);
16         }
17         while(n > 0);
18     }
19     public static double getDub(Scanner scan){
20         return scan.nextDouble(); }

```


Lehigh Username (use upper case letters): @lehigh.edu

total score

3) What is printed when the method q10() below is called? (20 points)

```

public static void q10() {
    int x[]={1,2,3,4,5}, y[]={7,8,9,10};
    conundrum1('a',4);
    conundrum1('b',2);
    conundrum1('c',3);
    print(x);
    print(y);
    conundrum2(x[1],x,y);
    print(x);
    print(y);
}

public static void print(int [] x){
    for(int j=0;j<x.length;j++)
        System.out.print(x[j]+" ");
    System.out.println();
}

public static void conundrum1(char ch,int n){
    switch(ch){
        case 'a':
            while(true){
                System.out.print(n+" ");
                n--;
                if(n<=0){
                    break;
                }
            }
        case 'b':
            System.out.println("At 'b'");
    }
}

public static void conundrum2(int x, int[] y, int [] z){
    int [] temp = y;
    x += 2;
    y = z;
    z[1] += 5;
    System.out.println(x);
    z = temp;
    z[3] = 42;
}

```

Your answer:

Lehigh Username (use upper case letters): @lehigh.edu

total score

4) Write a method called `stringToChars()`, which takes as input a `String`, and returns an **array of char**. The members of the output array should be same as the characters of the input string, and in the same order, except that the characters ' ' (space) and '.' (period) should be omitted. For example, `stringToChars("this is a string")` should return `"thisisastring"` as an array of chars.

You may use the `String` method `charAt()`, which returns the character at the given position, starting at location 0. For example, the following code prints out "h":

```
String ans = "This is a string. It is full of chars.";
System.out.println( charAt(1) ); (20 points)
```

Your answer:

Lehigh Username (use upper case letters): @lehigh.edu

total score

5) The method q9() below produces the indicated output, calling the methods starry() and print(). The method starry() returns an array of characters that the method print then displays. Write the code for starry(). Note: The char '0' is not the same as the int 0, but (char)('0'+0) is '0', (char)('0'+1) is '1', etc; (30 points)

```
public static void q9(){
    char [][] stars;
    stars=starry(5);
    print(stars);
    stars=starry(10);
    print(stars);
}
```

```
*000*
1*1*1
22*22
3*3*3
*444*
*00000000*
1*111111*1
22*2222*22
333*33*333
4444**4444
5555**5555
666*66*666
77*7777*77
8*888888*8
*99999999*
```

Your answer:

Lehigh Username (use upper case letters): @lehigh.edu

total score

6) Write a method, `inArray()`, that takes an int **k** and a two-dimensional array **A** of ints of arbitrary size, and returns true if and only if one member array of **A** contains **k**. In addition, the array **A** has the special properties that (a) every member array is sorted in ascending order and that (b) all members of one member array (i.e. **A**[i]) are smaller than the members of the next member array (i.e. **A**[i+1]). Assume **A** is initialized. **Use linear search to determine the member array where the given int might be. Use binary search in that member array to determine whether it contains the given int.** (20 points)

Lehigh Username (use upper case letters): @lehigh.edu

total score

7) Write a method q6() that takes two ints j and k for input and continually prompts the user until the user enters input that satisfies two conditions: it must be (a) an int **and** (b) between j and k, inclusive. Once the method q6() gets satisfactory input from the user, it should print the input and exit. If j>k, then q6 should print “this range is impossible”. Example: the q6(2,7) would prompt the user for an int between 2 and 7 and, if the user entered 2, it would display “You entered 2” and exit. If the user entered 9, q6 would prompt again. Assume that java.util.Scanner has been imported. You may use the Scanner method next(), which gets the next string of characters, up to a space or newline (20 points)

total score

Lehigh Username (use upper case letters): @lehigh.edu

8) Rewrite the code for the method allPos() below three times, once with for loops and once with while loops on this page, and once with do-while loops on the next page. (10 points each, 30 points total)

```
public static boolean allPos(int g[][]){
    for(int[] u: g){ // select each row of g
        for(int v: u){ //select each element of a row
            if(v < 0){
                return false;
            }
        }
    }
    return true;
}
```

For loops:**While loops:**

Lehigh Username (use upper case letters): @lehigh.edu

Do-While loops:

Lehigh Username (use upper case letters): @lehigh.edu

total score

9) Write a method called **ordered()** that takes as input a two-dimensional array of Strings and returns true if and only if **(a)** the strings in each member array are ordered by ascending length, **and (b)** that for all strings in one member array (i.e. $A[i]$) are shorter than the members of the next member array (i.e. $A[i+1]$). **If both properties do not hold, then ordered() returns false.** For example

```
String z[][]={{ "a","b","c","de"}, {"an","two","ton"}, {"three","fourth"}};  
String u[][]={ {"a","b","c","de"}, {"a","an","two","ton"}, {"three","fourth"}};
```

ordered(z) will return true, and ordered(u) will return false. (30 Points)

Lehigh Username (use upper case letters): @lehigh.edu

total score

10) An expanding array is a common feature of high level languages that is easily implemented in Java. Write the method, `doubleUp()`, which takes as input an array of ints and returns an array twice as long as the input array. The first half of the new array is to contain a insertion sorted copy of the data from the input array, and the rest is uninitialized. (20 points)