Name (print): __________________________________________

- **INSTRUCTIONS:**
  - Keep your eyes on your own paper and do your best to prevent anyone else from seeing your work.
  - Do NOT communicate with anyone other than the professor/proctor for ANY reason in ANY language in ANY manner.
  - This exam is closed notes, closed books, and no calculator.
  - Turn all mobile devices off and put them away now. You cannot have them on your desk.
  - Write neatly and clearly indicate your answers. What I cannot read, I will assume to be incorrect.
  - Stop writing when told to do so at the end of the exam. I will take 5 points off your exam if I have to tell you multiple times.
  - Academic misconduct will not be tolerated. Suspected academic misconduct will be immediately referred to the Emory Honor Council. Penalties for misconduct will be a zero on this exam, an F grade in the course, and/or other disciplinary action that may be applied by the Emory Honor Council.

- **TIME:** This exam has 6 questions on 9 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

---

*I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Emory community. I have also read and understand the requirements and policies outlined above.*

Signature: __________________________________________

<table>
<thead>
<tr>
<th>Question:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
1. (10 points) For each entry below, state whether the code is correct as written or has an error. If there is no error, write the output. If there is, show where the error is (you can circle the incorrect code) and explain why it is an error.

<table>
<thead>
<tr>
<th>Code</th>
<th>Error?</th>
<th>Output or Why is error?</th>
</tr>
</thead>
<tbody>
<tr>
<td>int[] A={1,2}; int k = A[3];</td>
<td>Yes</td>
<td>Length of array A is 2. Index 3 is out of bounds</td>
</tr>
<tr>
<td>String s1 = &quot;Strickland&quot;; s1.charAt(5)=’t’; System.out.println(s1);</td>
<td>Yes</td>
<td>charAt returns a char and cannot assign a char to a String</td>
</tr>
<tr>
<td>String s2 = &quot;Clara&quot;; int n = s2.length-1; System.out.println(n);</td>
<td>Yes</td>
<td>it should be s2.length()</td>
</tr>
<tr>
<td>String[] s1 = {&quot;Chuck&quot;, &quot;Berry&quot;}; char c1 = s1[1].charAt(1); System.out.println(c1);</td>
<td>No</td>
<td>'e'</td>
</tr>
<tr>
<td>String a = &quot;mi&quot;; int b = 88; System.out.print(b+a+&quot;/hr&quot;);</td>
<td>No</td>
<td>88mi/hr</td>
</tr>
<tr>
<td>public static int DMC12(boolean a){ if(a&gt;5) return true; return false; }</td>
<td>Yes</td>
<td>Input parameter of method is boolean, not int Return parameter is int, not boolean</td>
</tr>
<tr>
<td>public static void main(String[] args){ System.out.println(DMC12(10)); }</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. (15 points) Draw the array $A$ that would result after the following code is executed:

1. 
   ```java
   int[] A={9, 4, 3, 7, 2};
   for(int i=0; i<A.length; i++)
       A[i]=A[A.length-i-1];
   ```

   Answer:
   $\{2, 7, 3, 7, 2\}$

   Common mistake: Not noticing that the array is also changing

2. 
   ```java
   char[] B = {'a', 'b', 'c', 'd'};
   char[] A = new char[B.length];
   for(int i=0; i<B.length; i++)
       int index=(i+1)%B.length;
       A[i]=B[index];
   ```

   Answer:
   $\{'b','c','d','a'\}$

3. 
   ```java
   double[] A={4, 3, 8, 6, 9, 0};
   for (int i=0; i<A.length; i++)
       if(A[i] % 2 ==0)
           A[i]--;
       else
           A[i] +=2;
   ```

   Answer:
   $\{3,5,7,5,11,-1\}$
3. (15 points) Consider the following program:

```java
public class Flux{
    public static int marty = 3;
    public static double doc = 4.0;
    public static String biff = "Tannen";

    public static void hoverboard(int m, double d){
        m = (m+(int)d)/2;
        d=7.0;
        System.out.println(Flux.biff);  //<- Position 3
        Flux.biff = "Griff";
    }

    public static void main(String[] args){
        System.out.println(marty);  //<- Position 1
        System.out.println(Flux.marty);  //<- Position 2
        int marty = 2;
        hoverboard(marty,doc);
        System.out.println(marty);  //<- Position 4
        System.out.println(doc);  //<- Position 5
        System.out.println(biff);  //<- Position 6
        System.out.println(Flux.biff);  //<- Position 7
        {
            int mcFly = 8;
            System.out.println(marty);  //<- Position 8
        }
        System.out.println(mcFly);  //<- Position 9
        System.out.println(Flux.mcFly);  //<- Position 10
    }
}
```

1. What type of variable(s) does `hoverboard` return (if any)?
   No variable is returned.

2. What type of variable(s) does `hoverboard` take as an input?
   int, double
3. What is (are) the class variable(s)?

    int marty, double doc, String biff

4. What is (are) the local variable(s) in the main method?

    int marty, int mcFly

5. What is the name of the class?

    Flux

6. Determine what would be printed in the following positions. If a variable cannot be printed, write ERROR!
   - Position 1: 3
   - Position 2: 3
   - Position 3: ”Tannen”
   - Position 4: 2
   - Position 5: 4.0
   - Position 6: ”Griff”
   - Position 7: ”Griff”
   - Position 8: 2
   - Position 9: ERROR!
   - Position 10: ERROR!
4. (10 points) Give the output of the following program. Assume it compiles as written.

```java
public class Copernicus {

    public static void bttf(int[] arr){
        System.out.print("Jules") ;
        int[] r = new int[arr.length];
        int value = 8;
        r[0]=value;
        arr[0] = 7;
    }

    public static int bttf ( int [] arr , int value ) {
        System.out.print("Verne");
        int[] r = new int[arr.length];
        r[0]=value;
        value = 10;
        arr[0]=6;
        return 3;
    }

    public static void main(String[] args){
        int[] a = {7,10,14};
        int[] b = a;

        int value = 4;
        a[0]=value;
        System.out.println(b[0]);
        int n = bttf(a,value);
        System.out.println(value);
        System.out.println(a[0]);
        System.out.println(n);
    }
}
```

Output: 4, "Verne", 4, 6, 3
5. (30 points) Write the following methods:

1. Write a boolean method called `isElement` that, given an array of integers `A` and an integer `k`, returns true if `k` is in `A`, or false otherwise. For example, `isElement({3,4,5},4)` returns true, and `isElement({3,4,5},6)` returns false.

   ```java
   public static boolean isElement(int[] A, int k) {
       for (int i = 0; i < A.length; i++)
           if (A[i] == k)
               return true;
       return false;
   }
   ```

2. Write an int method called `different` that, given an array of integers `A`, determines the number of distinct (different) values in the array. For example, `different({4,2,3,2,2})` returns 3, and `different({5,5,5,5,5})` returns 1. (Hint: it is easier to determine how many elements are repeated, and then from the total length, subtract this number).

   ```java
   public static int different(int[] A) {
       int repetitions = 0;
       for (int i = 0; i < A.length - 1; i++)
           for (int j = i + 1; j < A.length; j++)
               if (A[i] == A[j]) {
                   repetitions++;
                   break;
               }
       return (A.length - repetitions);
   }
   ```

Common mistakes: 1) Setting the first counter to `A.length` (it crashes because `j` is asking for `i+1`). 2) Not breaking in the if statement. This causes numbers that are repeated more than once to be counted more than once as well. 3) returning "repetitions" instead of `A.length-repetitions`.

3. Write a method called `unique` that, given an array of integers `A`, returns a new array of integers with only the unique elements of `A`. For example, `unique({1,3,5,3,1})`
returns \{1,3,5\}, and unique(\{1,2,3\}) returns \{1,2,3\}. The order of the returned array is not important. You can use the methods you wrote in 1) and 2) and assume they work as described in the problem.

```java
public static int[] unique(int[] A) {
    int[] B = new int[unique.count(A)];
    int index = 0;
    for (int i = 0; i < A.length; i++) {
        if (!isElement(B, A[i])) {
            B[index++] = A[i];
        }
    }
    return B;
}
```
6. (20 points) You are given a 2D array of double values called \( \mathbf{A} \). Write a method called \texttt{DeLorean} that returns a 1D array with the sum of the elements of \( \mathbf{A} \) by rows. For example if \( \mathbf{A} \) is
\[
\begin{bmatrix}
1 & 3 & 8 \\
2 & 4 & 9
\end{bmatrix}
\]
the method should return \{12, 15\}

```java
public static double[] DeLorean(double[][] A) {
    int nrows = A.length, ncols = A[0].length;
    double[] B = new double[nrows];
    for (int row = 0; i < nrows; row++)
        for (int col = 0; j < ncols; col++)
            B[row] += A[row][col];

    return B;
}
```