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 9 questions on 14 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>7</th>
<th>8</th>
<th>9</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points</td>
<td>12</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>12</td>
<td>12</td>
<td>14</td>
<td>14</td>
<td>18</td>
<td>100</td>
</tr>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Describe what function the ‘+’ operator performs in the following statements; and evaluate the expression.

(a) (4 points) Given that: \texttt{int} \ a = 5, \ b = 3; 
\texttt{a + b}

\textbf{Solution:} Arithmetic Operators (Additive operator), return the sum of the two operands. Returns \texttt{int}, 8.

(b) (4 points) \texttt{’a’ + 1}

\textbf{Solution:} Arithmetic Operators (Additive operator), the character \texttt{a} will be casted/promoted to integer. Returns \texttt{int}, 98.

(c) (4 points) \texttt{"a" + "b"}

\textbf{Solution:} concatenate/join two strings ”a” and ”b”. Returns String ”ab”.

2. Describe what function the operators containing the symbol ’=’ perform in the following statements.

Given that: \texttt{int} \ a = 1, \ b = 2;

(a) (2 points) \texttt{a += b}

\textbf{Solution:} Assignment operator, equivalent to \texttt{a = a + b}; increment \texttt{a} on itself by \texttt{b}; \texttt{a} is 3 after this statement.

(b) (2 points) \texttt{a == b}

\textbf{Solution:} Equality operator, returns a boolean value; This expression returns false because \texttt{a} is equal to \texttt{b}.
3. (6 points) Convert the following binary numbers to decimal equivalents. (Show your work for partial credit)
   (a) $1101_2$
       \[13\]  
   (b) $111010_2$
       \[58\]

4. (8 points) Convert the following decimal number to their binary. (Show your work for partial credit)
   (a) 57
       \[111001\]
   (b) 47
       \[101111\]
5. (12 points) Evaluate each expression. Then give the result of the evaluation and the
data type of the result. If the expression cannot be evaluated or is not proper Java
syntax, you may simply write “error” for the value. The first row has been done for you.

```java
int i = 7, j = 4;
char c1 = 'a', c2 = 'B';
double d1 = 6.0, d2 = 3.0;
String s1 = "veni", s2="vidi", s3="vici";
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>j % i</td>
<td>4</td>
<td>int</td>
</tr>
<tr>
<td>j + d1</td>
<td>10.0</td>
<td>double</td>
</tr>
<tr>
<td>d1 + (int)d2</td>
<td>9.0</td>
<td>double</td>
</tr>
<tr>
<td>i + d2 &lt; j + d1</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>i-- != ++j + 1</td>
<td>true</td>
<td>boolean</td>
</tr>
<tr>
<td>c2 + 1</td>
<td>67</td>
<td>int</td>
</tr>
<tr>
<td>c1 &lt; c2</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>c1 - '1' == '0'</td>
<td>true</td>
<td>boolean</td>
</tr>
<tr>
<td>s1 + (c1 + 1)</td>
<td>&quot;veni98&quot;</td>
<td>String</td>
</tr>
<tr>
<td>'1' + c2 + &quot;&quot; + s2</td>
<td>&quot;115vidi&quot;</td>
<td>String</td>
</tr>
<tr>
<td>s3 + &quot;c1&quot;</td>
<td>&quot;vici1&quot;</td>
<td>String</td>
</tr>
<tr>
<td>s2.charAt(1) == s3.charAt(3)</td>
<td>true</td>
<td>boolean</td>
</tr>
<tr>
<td>s3.charAt(2)+1.0 == s2.charAt(2)</td>
<td>true</td>
<td>boolean</td>
</tr>
</tbody>
</table>

Notes:

- The parenthesis operator has the highest precedence of all operators.
- The logical not operator and the casting operator have higher precedence than
  arithmetic operators, relational/comparison operators, and the other logical oper-
  ators.
- Arithmetic operators have higher precedence than relational/comparison or logical
  operators.
- Relational/comparison operators have higher precedence than logical operators.
- Assignment operators have the lowest precedence of all operators
6. In the Emory Clinic we have 2 doctors on duty for student’s health issue. They’re Dr. Watson and Dr. Crick. Here is a program to query if they’re available based on their schedules. Suppose this program compiles and runs without error.
We use 1-7 to represent Sun, Mon, ..., Sat
We use 0 - 23 to represent 24 hours in a day (0=midnight, 12=noon).

```java
class Clinic {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Please input a day of week in integer(1-7): ");
        int day = in.nextInt();
        System.out.print("please input a time of day in integer(0-23): ");
        int time = in.nextInt();
        boolean watson=false;
        boolean crick=false;
        switch ( day ) {
            case 2:
                case 4:
                    case 6:
                        if ( time >=9 && time <=20 )
                            watson = true;
                        if (time >= 5 && time <=11 || time >= 18 && time <=23)
                            crick = true;
                        break;
            case 3:
            case 5:
                if ( time >=5 && time <=11 )
                    watson = true;
                if ( time >= 13 && time <=20)
                    crick = true;
                break;
            case 1:
            case 7:
                if (time >=6 && time <=9 || time >= 20 && time <= 22 )
                    watson = true;
                if (time >=15 && time <=20 )
                    crick = true;
                break;
        }
    }
}
```

default:
    System.out.println("invalid day!");
    break;
}

System.out.println("On day "+day+" at the time "+time+":");
if(watson)
    System.out.println("Dr. Watson is available!");
if(crick)
    System.out.println("Dr. Crick is available!");
if(!(watson || crick))
    System.out.println("no doctor available at this time!");
}
}

(a) (6 points) Suppose we want to check if any doctor is available on Tuesday morning, 11am.
Write down the output result if we enter 3 for day and 11 for time

Solution:
On day 3 at the time 11:
Dr. Watson is available!
grading:
2 points for the output format first message
4 points for the correctness of workflow

(b) (6 points) Suppose we want to check if any doctor is available on Saturday afternoon, 8pm.
Write down the output result if we enter 7 for day and 20 for time

Solution:
On day 7 at the time 20:
Dr. Watson is available!
Dr. Crick is available!
grading:
2 points for the output format first message
4 points for the correctness of workflow
7. If letters "A, B, C, ..., Z" are represented as numbers 1, 2, 3, ... 26, then

K-N-O-W-L-E-D-G-E
11+14+15+23+12+5+4+7+5 = 96
H-A-R-D-W-O-R-K
8+1+18+4+23+15+18+11 = 98
A-T-T-I-T-U-D-E
1+20+20+9+20+21+4+5=100

We want to find other words that equal to 100. Here we have a java program, which read in a word from user, then return the value of this word based on the above transforming rule. If user enters characters other than alphabet, such as numbers or symbols, the program will take only characters before the wrong character.

```java
import java.util.Scanner;
public class Attitude {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);

        System.out.print("Please enter a word: ");
        String word = myScanner.next();

        int res = 0;
        int i = 0;
        boolean valid = true;

        while(i < word.length() & valid){
            char c = word.charAt(i);
            if (c >= 'A' & c <= 'Z'){
                res += c - 64;
            }else if (c >= 'a' & c <= 'z'){
                res += c - 96;
            }else{
                valid = false;
                System.out.println("Invalid letter detected: " + c);
                String newWord = word.substring(0, i);
                System.out.println("The word \""+newWord+"\" equals "+res);
            }
            i++;
        }
        if (i == word.length()){
            System.out.println("The word \""+word+"\" equals "+res);
        }
    }
}
```
(a) (6 points) Write down the output result if we enter the word "hard"

Solution:
The word "hard" equals 31
grading:
2 points for the output format first message
4 points for the correctness of value

(b) (8 points) Write down the output result if we enter the word "h@rd"

Solution:
Invalid letter detected: @
The word "h" equals 8
grading:
2 points for the output format first message
2 points for the output format second message
4 points for the correctness of value
8. (14 points) Find the greatest common divisor of 2 integers. Your program should read 2 integers from the user, and then print out the largest common divisors of these two integers. Don’t forget that the range of common divisor will be from 1 to the smaller one between the two integers.

Examples of running this program:

```java
>>> java GCD
Please input the first integer:8
Please input the second integer:12
The Greatest Common Divisor of 8 and 12 is: 4

>>> java GCD
Please input the first integer:3
Please input the second integer:7
The Greatest Common Divisor of 3 and 7 is: 1
```

```java
import java.util.Scanner;
public class GCD {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Please input the first integer:");
        int a = in.nextInt();
        System.out.print("Please input the second integer:");
        int b = in.nextInt();
        /*------ Please Write Your Code Below This Line -------*/
    }
}
```
Solution:

```c
int i=1;
int min=a;
if (min>b){
    min=b;
}
```
int gcd=1;
while(i<=min){
    if( a%i==0 && b%i==0){
        gcd=i;
    }
    i++;
}
System.out.println("The Greatest Common Divisor of "+a+" and "+b+" is: "+gcd);

grading:
2 points: use variables correctly
2 points: find the min of a and b
2 points: initialize gcd outside loop
4 points: loop starts and terminates correctly
2 points: conditions on common divisor
2 points: output
9. (18 points) A DNA sequence is composed of 4 types of character: A, C, G and T. A CG site is a region where a 'C' is directly before of 'G', and it is of important functions. For example, the sequence "ATCGCAATTCGTTTCG" contains 3 CG site. You want to write a program to help computational biologists identify the number of CG sites in a given DNA string. Your program should read one string, and then count how many times the two characters "CG" appears in it.

Hint: You can assume the user-input string contains only A, C, G and T (no need to have proof-reading functions in your program). Your program will handle only single strand of DNA (no need to handle another complementary string of DNA).

Examples of running this program:

```java
>>> java CGFinder
Please enter a DNA sequence: ACGTACGT
There are 2 CG site(s) in the DNA sequence.

>>> java CGFinder
Please enter a DNA sequence: AGGGCTT
There are 0 CG site(s) in the DNA sequence.
```

```java
import java.util.Scanner;
public class CGFinder {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        System.out.print("Please enter a DNA sequence: ");
        String s = myScanner.next();
        /*----- Please Write Your Code Below This Line ------*/
```
Solution:

```java
int i=0;
int nCG=0;
while(i<s.length()-1){
  if(s.charAt(i)=='C'){
    if(s.charAt(i+1)=='G'){
      nCG++;
    }
  }
}
```
i++;
}
System.out.println("There're "+nCG+" CG site(s) in the DNA sequence.");

grading:
3 points: variable initialization
3 points: i start with 0
3 points: i+1 not going beyond boundary
3 points: counter increment condition correct
3 points: correct increment of the counter
3 points: output, format, etc.