# CS 46A Fall 2022

## Homework 05

### Requirements

1. You must name your classes exactly as specified. Otherwise Codecheck will not be able to process your submission and you will get no credit.
2. When you are finished with your code, submit it to Codecheck one final time then download the .signed.zip file.
3. You must upload all three signed.zip files together to Canvas and you should double check the files in Canvas to make sure all three zip files are uploaded.
4. Do not open the downloaded zip files. The files are digitally signed, and the grader program will check that they have not been opened.
5. Due time: 10 pm, Saturday, Oct 01. Submissions before the due time are not late.

   You will lose five points if your submission is marked Late in Canvas.

6. Grace time: 10 am, Sunday, Oct 02. Submissions before the grace time will not be rejected.

   You will receive no points if your submission is rejected by Canvas.

### Remember to follow our Programming Style Requirements!

### Problem 5A

We will do the Java application **`IntegerAndDouble`** from homework04 again with some additional features.

After prompting for the integer, your program must check the next input token: If it is not an integer, then display a message and terminate the program. Do the same for the double number.

You must use one Scanner object and call Scanner methods to process input. You cannot use the try-catch statement or call any parse method.

Before calculating the square root, your program must check the double number: If it is negative, then display a message; otherwise, calculate and display the square root.

Before calculating the quotient divided by the double number, your program must check the double number: If it is zero, then display a message; otherwise, calculate and display the quotient.

Before calculating the quotients and remainder divided by the integer part of the double number, your program must check the integer part of the double number: If it is zero, then display a message; otherwise, calculate and display the quotients and remainder.

The square root is displayed with 4 decimal digits, the quotient divided by the double number is displayed with 3 decimal digits, and the quotient divided by the integer part of the double number is displayed with 2 decimal digits.

Sample output I

```
Enter an integer: 5a
5a is not an integer!
Program terminated!
```

Sample output II

```
Enter an integer: 5
The integer number is 5.
Enter a double number:  homework05
homework05 is not a double number!
Program terminated!
```

Sample output III

```
Enter an integer: 5
The integer number is 5.
Enter a double number:  -3.3
The double number is -3.3.
No square root, since the double number is negative.
The quotient of the integer number divided by the double number is -1.515.
The integer part of the double number is -3.
The double quotient of the integer number divided by the integer part of the
double number is -1.67.
The integer quotient of the integer number divided by the integer part of the
double number is -1.
The remainder of the integer number divided by the integer part of the double
number is 2.
```

Sample output IV

```
Enter an integer: 21
The integer number is 21.
Enter a double number:  0.5
The double number is 0.5.
The square root of the double number is 0.7071.
The quotient of the integer number divided by the double number is 42.000.
The integer part of the double number is 0.
No quotient or remainder, since the integer part of the double number is zero.
```

Codecheck link for Problem 5A

## Problem 5B

Now for a bit of silliness.

According to the web comic xkcd, it is "creepy" to date someone too young. The comic uses the formula below to determine the lowest acceptable age for a date.

```
yourAge / 2 + 7
```

So, if your age is 24, the youngest person you should date is 19, and anyone below 19 is "creepy."

Your date also applies the same formula to his/her age to see if dating you would be "creepy." If your date is 22, then they should not date anyone younger than 18. You are 18 so you two are good, and you are off to a movie. But if your date is 26, dating anyone below 20 is "creepy" for them. You may want to rethink this date.

Write an application called **`DatingCreep`** to evaluate the creepiness of a date. Since it is an application, it will have a main() method. And we added one more condition on the age: the age must be between 18 and 50, inclusive, to use the program, both you and your date. Write the main() method according to the following pseudocode (assuming your name is Joe Clifton).

```
Display a prompt "What is your name? "
Read your name (name could have more than one word)
Display a prompt "What is your age? "
Read an int for your age
If your age is not in the required range
        Display the following message
                "You are too young or too old for our dating program."
                "Goodbye, Joe Clifton!"
        Terminate the program
Display a prompt "What is your date's age? "
Read an int for your date's age
If your date's age is in the required range
        Calculate and display your creep date age
        Calculate and display your date's creep date age
        If you are too young
                Display message "Creepy alert, Joe Clifton! You are too young for your date!"
        Else if your date is too young
                Display message "Creepy alert, Joe Clifton! Your date is too young for you!"
        Else
                Display message "Have a good time, Joe Clifton!"
Otherwise
        Display the following message
                "Your date is too young or too old for our dating program."
                "Goodbye, Joe Clifton!"
```

You must use one Scanner object and call Scanner methods to process input. You cannot use the try-catch statement or call any parse method.

You must declare constants for all numbers used in the program, including number 2. No "magic numbers!"

Sample output I

```
What is your name? Joe Clifton
What is your age? 22
What is your date's age? 18
Your creepy age is 18.
Your date's creepy age is 16.
Have a good time, Joe Clifton!
```

Sample output II

```
What is your name? Jon
What is your age? 22
What is your date's age? 17
Your date is too young or too old for our dating program.
Goodbye, Jon!
```

Sample output III

```
What is your name? Steve Young
What is your age? 30
What is your date's age? 21
Your creepy age is 22.
Your date's creepy age is 17.
Creepy alert, Steve Young! Your date is too young for you!
```

Codecheck link for Problem 5B

## Problem 5C

Write a class called **`MyNumber`** that represents a non-negative integer.

Provide the constructor:

- **`public MyNumber(int value)`** If *value* is negative, multiply if by -1 so that you do not have a negative number. You should call method setNumber() and do not write the validity check code twice.

Provide methods:

- **`public int getNumber()`** Gets this number.

- `public void setNumber(int value)` Sets the stored number to a new value. If *value* is negative, multiply if by -1 so that you do not have a negative number.
- `public int digitCount()` Gets the number of digits in this number.
- `public String formatWithComma()` Gets a string representing the number in the comma format, if the number is less than 1,000,000. If the number is greater than or equal to 1,000,000, return the string "too big".

Hint: You can convert a number to a string by concatenating it with the empty string.

You must define constants for all numbers used in the program, except 0.

Codecheck link for Problem 5C