Assignment 3

- Submit all the deliverables to CourSys: https://coursys.sfu.ca
- This assignment is to be **done individually**. Do not show another student your code, do not copy code found online or from previous course offerings, and do not post questions about the assignment to online forums. Please direct all questions to the instructor or TA(s).
- See the marking guide for details on how each part will be marked.

1. BOGUS CO_2 to Temperature Table[1]

It is well established (though surprisingly commonly doubted) that the level of atmospheric CO_2 contributes to the greenhouse effect which warms the planet. Scientists study the effect of CO_2 (and other greenhouse gases) on climate change using computers to model the planet and predict the effects of different levels of emissions. These systems are exceptionally complicated and modeled by very sophisticated programs.

The problem is real, but *the math in this assignment is not*!

Imagine you work at BOGUS Science Inc, and your job is to make a table that uses many of the *same words that real scientists use*! Here are the assumptions we will be working under:

- The current year is 2024, with a CO_2 level of 419.3 parts per million (ppm)[2]
- The planet's average temperature will rise (cool) by 0.0091°C for each additional CO_2 ppm added (removed) from the atmosphere. So if CO_2 ppm rises by 100, it will increase the average temperature by 0.91°C. Reducing CO_2 ppm by 100 will decrease the average temperature by 0.91°C.[3]
- Vancouver's average daytime high temperature during the summer is 22°C.[4]

Write the program `co2.cpp` that prints a table of CO_2 levels and average temperature changes for a number of years.

- The program asks the user for:
  - The yearly change in CO_2 (in ppm). May be positive or negative. Will be a value such as 2.5, or -0.01352.
  - The number of years to print the table for.
- You do not need to do any error checking on the user's input; assume they correctly enter numbers, and that the numbers are fine. For example, you do *not* have to check if the number of years is positive.

1 Global warming is complicated! Have a look at the IPCC report https://www.ipcc.ch/sr15/
2 This is reasonably true: https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide
3 This is **bogus**, **contrived**, **fake**, and **phony**: https://www.thesaurus.com/browse/bogus
4 Value from https://en.wikipedia.org/wiki/Climate_of_Vancouver

- Print a table to the screen featuring the following columns:
  - Year
  - CO_2 (in ppm)
  - Change in temperature for this year over the initial temperature (in °C)
  - Average daytime high temperature in Vancouver during the summer, assuming Vancouver's temperature changes at the same rate as the planet overall (in °C)
  - Comment on the change in average temperature from the initial temperature:

Change in temperature °C	**Comment**
Drop by -1°C or more	`Getting cold!`
Change (-1.0, -0.5] i.e.: -1.0 < change <= -0.5	`Getting cooler.`
Change (-0.5, +0.5)	<No message>
Change [+0.5, +1.0)	`Getting warmer.`
Change [+1.0, +1.5)	`That's the dream!`
Change [+1.5, +2.0)	`Think we can hold it here?`
Change [+2.0, +5.0)	`Uh oh! It's HOT!`
Rise by +5.0°C or more	<Write your own message!>

- The table must be nicely formatted.
  - The temperature change column must show either a + or -. To do this, use the identifiers `showpos` and `noshowpos` defined in `iostream` code like:

```
cout << showpos << 3.5 << noshowpos;
```

  - The first year has no change to the initial CO_2 ppm or temperature (it shows the baseline).
  - All temperatures and ppm must be shown to 2 decimal places.
  - *Hints*
    - *Align your values using `setw()`, and then print the units after the value.*
    - *Do the same for the headings: use `setw()` for the heading, then print the units next.*
    - *There's one column you don't need to use `setw()` for.*
    - *Make constants for widths.*
    - *Use `fixed` and `setprecision()` for getting decimal places (in iomanip).*
- Code Style:
  - Use named constants appropriately: do not use any magic numbers.
  - Indent and format your code correctly.
  - Comment your code.
  - You may (but don't have to) use functions in your code.
- Suggestions:
  - Write the psedo-code first to figure out your algorithm.
  - Code in small pieces, making sure things work before adding more code.
  - Format the table last. Note there is a space between the total temperature change and the comment columns. In addition to allocating space for the number, also try printing an extra space between the columns.

- Sample Outputs:
  - User input shown in green italics; your user input need not be styled in any specific way.

```
BOGUS CO2 to Temperature Change Table Generator
-----------------------------------------------
Enter the yearly change in CO2 [ppm]: 20.3
Enter the number of years to print: 15

 Year         CO2ppm   Temp Chng'C  Van Summer'C  Comment
 2024      419.30ppm       +0.00'C       22.00'C
 2025      439.60ppm       +0.18'C       22.18'C
 2026      459.90ppm       +0.37'C       22.37'C
 2027      480.20ppm       +0.55'C       22.55'C  Getting warmer
 2028      500.50ppm       +0.74'C       22.74'C  Getting warmer
 2029      520.80ppm       +0.92'C       22.92'C  Getting warmer
 2030      541.10ppm       +1.11'C       23.11'C  That's the dream!
 2031      561.40ppm       +1.29'C       23.29'C  That's the dream!
 2032      581.70ppm       +1.48'C       23.48'C  That's the dream!
 2033      602.00ppm       +1.66'C       23.66'C  Think we can hold it here?
 2034      622.30ppm       +1.85'C       23.85'C  Think we can hold it here?
 2035      642.60ppm       +2.03'C       24.03'C  Uh oh! It's HOT!
 2036      662.90ppm       +2.22'C       24.22'C  Uh oh! It's HOT!
 2037      683.20ppm       +2.40'C       24.40'C  Uh oh! It's HOT!
 2038      703.50ppm       +2.59'C       24.59'C  Uh oh! It's HOT!
 2039      723.80ppm       +2.77'C       24.77'C  Uh oh! It's HOT!
```

```
BOGUS CO2 to Temperature Change Table Generator
-----------------------------------------------
Enter the yearly change in CO2 [ppm]: -18.6
Enter the number of years to print: 8

 Year         CO2ppm   Temp Chng'C  Van Summer'C  Comment
 2024      419.30ppm       +0.00'C       22.00'C
 2025      400.70ppm       -0.17'C       21.83'C
 2026      382.10ppm       -0.34'C       21.66'C
 2027      363.50ppm       -0.51'C       21.49'C  Getting cooler
 2028      344.90ppm       -0.68'C       21.32'C  Getting cooler
 2029      326.30ppm       -0.85'C       21.15'C  Getting cooler
 2030      307.70ppm       -1.02'C       20.98'C  Getting cold!
 2031      289.10ppm       -1.18'C       20.82'C  Getting cold!
 2032      270.50ppm       -1.35'C       20.65'C  Getting cold!
```

```
BOGUS CO2 to Temperature Change Table Generator
-----------------------------------------------
Enter the yearly change in CO2 [ppm]: 2.5
Enter the number of years to print: 25

 Year         CO2ppm   Temp Chng'C  Van Summer'C  Comment
 2024      419.30ppm       +0.00'C       22.00'C
 2025      421.80ppm       +0.02'C       22.02'C
 2026      424.30ppm       +0.05'C       22.05'C
 2027      426.80ppm       +0.07'C       22.07'C
 2028      429.30ppm       +0.09'C       22.09'C
 2029      431.80ppm       +0.11'C       22.11'C
 2030      434.30ppm       +0.14'C       22.14'C
 2031      436.80ppm       +0.16'C       22.16'C
 2032      439.30ppm       +0.18'C       22.18'C
 2033      441.80ppm       +0.20'C       22.20'C
 2034      444.30ppm       +0.23'C       22.23'C
 2035      446.80ppm       +0.25'C       22.25'C
 2036      449.30ppm       +0.27'C       22.27'C
 2037      451.80ppm       +0.30'C       22.30'C
 2038      454.30ppm       +0.32'C       22.32'C
 2039      456.80ppm       +0.34'C       22.34'C
 2040      459.30ppm       +0.36'C       22.36'C
 2041      461.80ppm       +0.39'C       22.39'C
 2042      464.30ppm       +0.41'C       22.41'C
 2043      466.80ppm       +0.43'C       22.43'C
 2044      469.30ppm       +0.46'C       22.45'C
 2045      471.80ppm       +0.48'C       22.48'C
 2046      474.30ppm       +0.50'C       22.50'C  Getting warmer
 2047      476.80ppm       +0.52'C       22.52'C  Getting warmer
 2048      479.30ppm       +0.55'C       22.55'C  Getting warmer
 2049      481.80ppm       +0.57'C       22.57'C  Getting warmer
```

NOTE: Earth's yearly change in CO_2 ppm is actually about +2.5.[5]

5 Source: https://climate.nasa.gov/climate_resources/296/global-carbon-dioxide-2020-2021/
However, the effect of this on temperatures is completely made up in this assignment!

2. Dice Game: Beat The Roll

Write a program named `beattheroll.cpp` which plays the Beat The Roll dice game described below. Note that this game is not a standard game; it was created just for this assignment.

2.1 Game Description

- There is only one player (the user), plus the dealer (the computer).
- The player starts with 50 points. He or she wins upon reaching 100 points (or more), but loses upon reaching 0 points.
- Each round starts with the dealer rolling two dice (and adds them together). The player can see the roll.
- The player then bets a certain number of points.
- The player then rolls two dice (and adds them together).
  - If the player beats the dealer (player's sum is greater than dealer's sum), then the player wins as many points as he or she bet.
  - If the player ties the dealer (player's sum equals the dealer's sum), then no points are won or lost.
  - If the dealer beats the player (player's sum is less than dealer's sum), then the player loses as many points as he or she bet.
- The game continues until the player wins (has 100 or more points) or loses (has 0 points).

2.2 Required Functions

- Your program must include the functions described below; you are free to create more functions than just these if you like.
- Before implementing the game, implement each of these functions and carefully test them.
- After they all work, you can then integrate them with your game.
- *Hint: Place each of these functions above the `main()` function to avoid having to use function prototypes for this assignment.*
  - *You may use prototypes if you wish, but you do not need to.*

2.2.1 Welcome Message

Create a function which prints a welcome message to the screen:

```
****************************************
Welcome to Roller's Un-Random house of dice!
****************************************
```

Test it by calling your function from the `main()` function.

2.2.2 Get the user's name

Create a well-named function which prompts the user to enter his or her first name. Make the function return the user's name.

```
What is your first name? Brian
```


2.2.3 Random: Seed

Create a function which allows us to seed the random number generator in two ways. It should ask the user how to seed the random number generator. If the user enters 0, then seed using the timer. If the user enters any number other than 0, use that value to seed the generator.
Hint: Have this function actually seed the random number generator by calling `srand()`. No other functions should call srand(). Your program will only ever call `srand()` once.
Sample outputs when called from `main()`:[6]

```
Would you like to pick an un-random game, or let the timer pick?
Enter 0 for timer, or pick your own un-random game: 0
THE TIMER! A daring choice!
```

```
Would you like to pick an un-random game, or let the timer pick?
Enter 0 for timer, or pick your own un-random game: 42
42! A wise and safe choice.
```

2.2.4 Random: Rolling

Create a well named function which *returns* a random number between 1 and 6 inclusive by using the `rand()`.
Hint: Test it function by calling it from `main()` using a loop! Once it's proven to work, remove this test code.

2.2.5 Get Max Bet

Create a function which asks the user what the betting limit will be for this game. The limit must be greater than or equal to 1. Have the function return this value so the calling code can use the value.

- This is the maximum number of points the user will be allowed to bet.
  - For example, the user could choose a low limit like 10, or a high limit like 100 or 200. There is no upper limit.
  - In "reality", this limit would be imposed by the casino.
- When the maximum bet value is too low, have it print an error message.

Test your function by calling it from `main()`. Verify it correctly enforces the constraints listed above. Sample outputs:

```
What would you like to be the maximum bet? :500
```

```
What would you like to be the maximum bet? :-1
The maximum bet must be greater than or equal to 1.
What would you like to be the maximum bet? :0
The maximum bet must be greater than or equal to 1.
What would you like to be the maximum bet? :60
```

6 This is done so that we can specify a specific seed, and hence test how the game reacts to certain winning/losing conditions. Normally, a game like this would just seed using the timer, but for marking we want to be able to test your program without the randomness!

2.2.6 Get User's Bet

Create a well named function which asks the user for their bet:

- Ask the user to enter their bet.
- Ensure that the user's bet is:
  - at least the minimum bet of 1 point;
  - no more than the maximum bet (entered by the user at the start of the program);
  - no more than the user's current number of points.
- If the bet is invalid, display an error message explaining why the bet is invalid and re-ask the user for the bet.
  - If the bet is greater than the maximum *and* greater than the user's score, either error message can be displayed.
- Design the function to accept, as parameters, the values that it needs to do this work.
- Design the function to return the user's bet so that it can be used by the rest of the program.

Test your function by calling it from `main()`. Verify it correctly enforces the constraints listed above. Sample outputs:

```
Enter your bet: 5
```

```
Enter your bet: 0
Your must bet at least 1.
Enter your bet: 30
Your must not bet more than the maximum bet (25).
Enter your bet: 100
Your must not bet more than your score (50).
Enter your bet: 12
```

Output 1: Sample output showing constraints when given user's current score of 50 and maximum bet of 25.

2.3 Program Description

Remove from `main()` any test code for testing your function before writing the game.

- Display a welcome message (call your function!)
- Get the user's name (call your function); store it in a local variable in `main()`.
- Setup the pseudo-random generator (call your function).
- Get the maximum bet (call your function).
  - *Hint: This function should returns the maximum bet, so store the return value in a variable for later use in your program.*
- The user plays rounds of the game (as described above) until he or she wins (>=100 points) or loses (0 points):
  - Get the rolls and user's bet by calling the functions you already wrote.
  - Each round, determine if the user...
    - won the round: then award points from the user;
    - lost the round: then subtract points from the user;
    - tied the round: then no points awarded or subtracted.
  - Only track the score of the user; do not track the score of the dealer.
  - *Hint: Write down the steps on paper that you would need to do if you were the dealer. Then mark which of these steps (or part of a step) can be done by your functions.*
- When the user wins or loses the game display an appropriate message and end the game.
  - Include the user's name in your win/loss message.

- As always, your code must have good style: meaningful comments and use named constant.
  - Note that often you won't need named constants for 0 or 1; however, for this program the minimum bet (1) and the losing score (0) should also be named constants because someone might very well want to change those values in the future.
- You may assume that the user enters the correct *type* of data when required.
  - You must ensure that the numbers the user enters are valid. For example, if a value must be at least 1, you must reject 0 and negative numbers by re-asking the user for a value (as shown in the second sample output below).
- Your output should be quite similar to the output shown below.
- Sample winning output:

```
*****************************************
Welcome to Roller's Un-Random house of dice!
*****************************************
What is your first name? Brian
Would you like to pick an un-random game, or let the timer pick?
Enter 0 for timer, or pick your own un-random game: 0
THE TIMER! A daring choice!

What would you like to be the maximum bet? :200

Round 1 You have 50 points.
Dealer rolls: 2 + 1 =  3                Enter your bet: 20
You roll:     5 + 4 =  9.
Brian, you won! :-)
Current score: 70.

Round 2 You have 70 points.
Dealer rolls: 3 + 5 =  8                Enter your bet: 2
You roll:     5 + 1 =  6.
Brian, you lost. :-(
Current score: 68.

Round 3 You have 68 points.
Dealer rolls: 4 + 4 =  8                Enter your bet: 60
You roll:     6 + 3 =  9.
Brian, you won! :-)
Current score: 128.
Congratulations Brian! You win the game with a score of 128.
```


- Sample losing output demonstrating some error checking:

```
****************************************
Welcome to Roller's Un-Random house of dice!
****************************************
What is your first name? Brian
Would you like to pick an un-random game, or let the timer pick?
Enter 0 for timer, or pick your own un-random game: 42
42! A wise and safe choice.

What would you like to be the maximum bet? :0
The maximum bet must be greater than or equal to 1.
What would you like to be the maximum bet? :-1
The maximum bet must be greater than or equal to 1.
What would you like to be the maximum bet? :40

Round 1 You have 50 points.
Dealer rolls: 2 + 5 =  7                Enter your bet: 45
Your must not bet more than the maximum bet (40).
Enter your bet: 25
You roll:     2 + 3 =  5.
Brian, you lost. :-(
Current score: 25.

Round 2 You have 25 points.
Dealer rolls: 4 + 4 =  8                Enter your bet: 20
You roll:     1 + 1 =  2.
Brian, you lost. :-(
Current score: 5.

Round 3 You have 5 points.
Dealer rolls: 6 + 6 = 12                Enter your bet: 0
Your must bet at least 1.
Enter your bet: -1
Your must bet at least 1.
Enter your bet: 5
You roll:     4 + 5 =  9.
Brian, you lost. :-(
Current score: 0.
I'm sorry, Brian; you are out of points so you lose.
```

3. Deliverables

Submit the following two files to the CourSys: https://coursys.sfu.ca

1. `co2.cpp`
2. `beattheroll.cpp`

Submit both files at once! If you submit them individually, we will only initially mark the final submission.

Please remember that all submissions will automatically be compared for unexplainable similarities. This comparison will also include similar assignments from previous semesters and programs on the internet. Please review the notes from lecture on the expectations for academic honesty.

