Assignment 3

Submit all the deliverables to CourSys: https://courses.cs.sfu.ca/

There is no possibility of turning the assignment in late because the solution will be posted soon after it is due in preparation for the midterm.

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 CO₂ to Temperature Table¹

It is well established (though surprisingly poorly accepted!) that the level of atmospheric CO₂ contributes to the greenhouse effect which warms the planet. Scientists study the effect of CO₂ (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 2019, with a CO₂ level of 366 parts per million (ppm)²
- The current average temperature for the planet is 14.5°C³
- The planet’s average temperature will rise (cool) by 0.0091°C for each additional CO₂ ppm added (removed) from the atmosphere. So if CO₂ ppm rises by 100, it will increase the average temperature by 0.91°C. Reducing CO2 ppm by 100 will decrease the average temperature by 0.91°C.⁴

Write the program co2.cpp that prints a table of CO₂ levels and average temperatures for a number of years.

The program asks the user for:
- The yearly change in CO2 (in ppm). May be positive or negative. Will be a value such as 2.2, 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 not too far off: https://www.currentresults.com/Environment-Facts/changes-in-earth-temperature.php
⁴ This is contrived, fake, and phony: https://www.thesaurus.com/browse/bogus
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)
- BOGUS average temperature (in °C)
- Comment on the change in average temperature from the initial temperature:

<table>
<thead>
<tr>
<th>Change in temperature °C</th>
<th>Comment</th>
<th>Row Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop by -1°C or more</td>
<td>Getting cold!</td>
<td>Bold blue</td>
</tr>
<tr>
<td>Change (-1.0, -0.5)</td>
<td>Getting cooler.</td>
<td>Blue</td>
</tr>
<tr>
<td>i.e.: -1.0 &lt; change &lt;= -0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change (-0.5, +0.5)</td>
<td>&lt;No message&gt;</td>
<td>Normal</td>
</tr>
<tr>
<td>Change [+0.5, +1.0)</td>
<td>Getting warmer.</td>
<td>Green</td>
</tr>
<tr>
<td>Change [+1.0, +1.5)</td>
<td>That’s the dream!</td>
<td>Yellow</td>
</tr>
<tr>
<td>Change [+1.5, +2.0)</td>
<td>Think we can hold it here?</td>
<td>Red</td>
</tr>
<tr>
<td>Change [+2.0, +5.0)</td>
<td>Uh oh! It’s HOT!</td>
<td>Bold red</td>
</tr>
<tr>
<td>Rise by +5.0°C or more</td>
<td>&lt;Write your own message!&gt;</td>
<td>Bold black text on a red background</td>
</tr>
</tbody>
</table>

The table must be nicely formatted.
- **Hint:** Use ~12 on-screen columns for each column of data. You need not do this exactly.
- **Hint:** Align your values using setw(), and then print the units after the value.
- **Hint:** Do the same trick 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.
- The temperature change column must show either a + or -. To do this, use code like:
  ```cpp
cout << showpos << 3.5 << noshowpos;
```
- The first year has no change to the initial CO\(_2\) ppm or temperature.
- **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.
- **Colors:**
  - The column titles in your table must be bold.
  - You must colour the rows of your table as listed in the above table for “Row Color”.
- **Suggestions:**
  - Write the pseudo-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 average temperature and the comment columns. In addition to allocating space for the number (using something like `setw(12)`), 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 Table Generator

Enter the yearly change in CO2 [ppm]: 20.3
Enter the number of years to print: 15

<table>
<thead>
<tr>
<th>Year</th>
<th>CO2ppm</th>
<th>Temp Chng*C</th>
<th>BOGUS Avg*C</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>366.00ppm</td>
<td>+0.00*C</td>
<td>14.50*C</td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td>386.30ppm</td>
<td>+0.18*C</td>
<td>14.68*C</td>
<td></td>
</tr>
<tr>
<td>2021</td>
<td>406.60ppm</td>
<td>+0.37*C</td>
<td>14.87*C</td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td>426.90ppm</td>
<td>+0.55*C</td>
<td>15.05*C</td>
<td>Getting warmer</td>
</tr>
<tr>
<td>2023</td>
<td>447.20ppm</td>
<td>+0.74*C</td>
<td>15.24*C</td>
<td>Getting warmer</td>
</tr>
<tr>
<td>2024</td>
<td>467.50ppm</td>
<td>+0.92*C</td>
<td>15.42*C</td>
<td>Getting warmer</td>
</tr>
<tr>
<td>2025</td>
<td>487.80ppm</td>
<td>+1.11*C</td>
<td>15.61*C</td>
<td>That's the dream!</td>
</tr>
<tr>
<td>2026</td>
<td>508.10ppm</td>
<td>+1.29*C</td>
<td>15.79*C</td>
<td>That's the dream!</td>
</tr>
<tr>
<td>2027</td>
<td>528.40ppm</td>
<td>+1.48*C</td>
<td>15.98*C</td>
<td>That's the dream!</td>
</tr>
<tr>
<td>2028</td>
<td>548.70ppm</td>
<td>+1.66*C</td>
<td>16.16*C</td>
<td>Think we can hold it here?</td>
</tr>
<tr>
<td>2029</td>
<td>569.00ppm</td>
<td>+1.85*C</td>
<td>16.35*C</td>
<td>Think we can hold it here?</td>
</tr>
<tr>
<td>2030</td>
<td>589.30ppm</td>
<td>+2.03*C</td>
<td>16.53*C</td>
<td>Uh oh! It's HOT!</td>
</tr>
<tr>
<td>2031</td>
<td>609.60ppm</td>
<td>+2.22*C</td>
<td>16.72*C</td>
<td>Uh oh! It's HOT!</td>
</tr>
<tr>
<td>2032</td>
<td>629.90ppm</td>
<td>+2.40*C</td>
<td>16.90*C</td>
<td>Uh oh! It's HOT!</td>
</tr>
<tr>
<td>2033</td>
<td>650.20ppm</td>
<td>+2.59*C</td>
<td>17.09*C</td>
<td>Uh oh! It's HOT!</td>
</tr>
<tr>
<td>2034</td>
<td>670.50ppm</td>
<td>+2.77*C</td>
<td>17.27*C</td>
<td>Uh oh! It's HOT!</td>
</tr>
</tbody>
</table>

BOGUS CO2 to Temperature Table Generator

Enter the yearly change in CO2 [ppm]: -18.6
Enter the number of years to print: 8

<table>
<thead>
<tr>
<th>Year</th>
<th>CO2ppm</th>
<th>Temp Chng*C</th>
<th>BOGUS Avg*C</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>366.00ppm</td>
<td>+0.00*C</td>
<td>14.50*C</td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td>347.40ppm</td>
<td>-0.17*C</td>
<td>14.33*C</td>
<td></td>
</tr>
<tr>
<td>2021</td>
<td>328.80ppm</td>
<td>-0.34*C</td>
<td>14.16*C</td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td>310.20ppm</td>
<td>-0.51*C</td>
<td>13.99*C</td>
<td>Getting cooler</td>
</tr>
<tr>
<td>2023</td>
<td>291.60ppm</td>
<td>-0.68*C</td>
<td>13.82*C</td>
<td>Getting cooler</td>
</tr>
<tr>
<td>2024</td>
<td>273.00ppm</td>
<td>-0.85*C</td>
<td>13.65*C</td>
<td>Getting cooler</td>
</tr>
<tr>
<td>2025</td>
<td>254.40ppm</td>
<td>-1.02*C</td>
<td>13.48*C</td>
<td>Getting cold!</td>
</tr>
<tr>
<td>2026</td>
<td>235.80ppm</td>
<td>-1.18*C</td>
<td>13.32*C</td>
<td>Getting cold!</td>
</tr>
<tr>
<td>2027</td>
<td>217.20ppm</td>
<td>-1.35*C</td>
<td>13.15*C</td>
<td>Getting cold!</td>
</tr>
</tbody>
</table>
Enter the yearly change in CO₂ [ppm]: **2.4**
Enter the number of years to print: **25**

<table>
<thead>
<tr>
<th>Year</th>
<th>CO₂ ppm</th>
<th>Temp Chng °C</th>
<th>BOGUS Avg °C</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>366.00ppm</td>
<td>+0.00°C</td>
<td>14.50°C</td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td>368.40ppm</td>
<td>+0.02°C</td>
<td>14.52°C</td>
<td></td>
</tr>
<tr>
<td>2021</td>
<td>370.80ppm</td>
<td>+0.04°C</td>
<td>14.54°C</td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td>373.20ppm</td>
<td>+0.07°C</td>
<td>14.57°C</td>
<td></td>
</tr>
<tr>
<td>2023</td>
<td>375.60ppm</td>
<td>+0.09°C</td>
<td>14.59°C</td>
<td></td>
</tr>
<tr>
<td>2024</td>
<td>378.00ppm</td>
<td>+0.11°C</td>
<td>14.61°C</td>
<td></td>
</tr>
<tr>
<td>2025</td>
<td>380.40ppm</td>
<td>+0.13°C</td>
<td>14.63°C</td>
<td></td>
</tr>
<tr>
<td>2026</td>
<td>382.80ppm</td>
<td>+0.15°C</td>
<td>14.65°C</td>
<td></td>
</tr>
<tr>
<td>2027</td>
<td>385.20ppm</td>
<td>+0.17°C</td>
<td>14.67°C</td>
<td></td>
</tr>
<tr>
<td>2028</td>
<td>387.60ppm</td>
<td>+0.20°C</td>
<td>14.70°C</td>
<td></td>
</tr>
<tr>
<td>2029</td>
<td>390.00ppm</td>
<td>+0.22°C</td>
<td>14.72°C</td>
<td></td>
</tr>
<tr>
<td>2030</td>
<td>392.40ppm</td>
<td>+0.24°C</td>
<td>14.74°C</td>
<td></td>
</tr>
<tr>
<td>2031</td>
<td>394.80ppm</td>
<td>+0.26°C</td>
<td>14.76°C</td>
<td></td>
</tr>
<tr>
<td>2032</td>
<td>397.20ppm</td>
<td>+0.28°C</td>
<td>14.78°C</td>
<td></td>
</tr>
<tr>
<td>2033</td>
<td>399.60ppm</td>
<td>+0.31°C</td>
<td>14.81°C</td>
<td></td>
</tr>
<tr>
<td>2034</td>
<td>402.00ppm</td>
<td>+0.33°C</td>
<td>14.83°C</td>
<td></td>
</tr>
<tr>
<td>2035</td>
<td>404.40ppm</td>
<td>+0.35°C</td>
<td>14.85°C</td>
<td></td>
</tr>
<tr>
<td>2036</td>
<td>406.80ppm</td>
<td>+0.37°C</td>
<td>14.87°C</td>
<td></td>
</tr>
<tr>
<td>2037</td>
<td>409.20ppm</td>
<td>+0.39°C</td>
<td>14.89°C</td>
<td></td>
</tr>
<tr>
<td>2038</td>
<td>411.60ppm</td>
<td>+0.41°C</td>
<td>14.91°C</td>
<td></td>
</tr>
<tr>
<td>2039</td>
<td>414.00ppm</td>
<td>+0.44°C</td>
<td>14.94°C</td>
<td></td>
</tr>
<tr>
<td>2040</td>
<td>416.40ppm</td>
<td>+0.46°C</td>
<td>14.96°C</td>
<td></td>
</tr>
<tr>
<td>2041</td>
<td>418.80ppm</td>
<td>+0.48°C</td>
<td>14.98°C</td>
<td></td>
</tr>
<tr>
<td>2042</td>
<td>421.20ppm</td>
<td>+0.50°C</td>
<td>15.00°C</td>
<td>Getting warmer</td>
</tr>
<tr>
<td>2043</td>
<td>423.60ppm</td>
<td>+0.52°C</td>
<td>15.02°C</td>
<td>Getting warmer</td>
</tr>
<tr>
<td>2044</td>
<td>426.00ppm</td>
<td>+0.55°C</td>
<td>15.05°C</td>
<td>Getting warmer</td>
</tr>
</tbody>
</table>

NOTE: Yearly change in CO₂ ppm is ~2.2 or 2.4.
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 named `displayWelcome()` which accepts no arguments and returns no value (`void`). Have it print the following 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 function named `getUserName()` which prompts the user to enter his or her first name. Have the function return the user's name as a string. Sample output when called from `main()`.

```
What is your first name? Brian
```
2.2.3 Random: Seed

Create a function named `seedGame()` which accepts no arguments and returns no value. 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: This function actually does 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()`:

```
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 function named `getRandomRoll()` which accepts no arguments and returns an `int`. It should return a random number between 1 and 6 inclusive by using the `rand()` function as shown in class. Test that your function works by calling it from `main()` about 20 times and printing the value to the screen. Once tested, remove these test calls from `main()`.  

*Hint: Test it using a loop!*

2.2.5 Get Max Bet

Create a function named `getMaxBet()` which accepts no arguments and returns an `int`. It asks the user what the betting limit will be for this game. The limit must be greater than or equal to 1.  

- 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 in red.

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
```

---

5 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 Bet
Create a function named `getUserBet()` which returns an integer (the bet), and accepts two integer parameters (the user's current score, and the maximum 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 in red and loop until valid.
- It is undefined which error message should be displayed if the bet is greater than the maximum and greater than the user's score. (i.e., you may show either message in this case).

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

```plaintext
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.

- Call your `displayWelcome()` to display the welcome message.
- Get the user's name by calling `getUserName()`; store the return value in a string.
- Setup the pseudo-random generator with a seed by calling your `seedGame()` function.
- Setup the maximum bet by calling `getMaxBet()`.
  
  **Hint:** This function returns the maximum bet, so store the return value in a variable for later use in your game.

- The user plays rounds of the game (as described above) until he or she wins (>=100 points) or loses (0 points):
  - All rolls are done by using pseudo-random numbers by calling `getRandomRoll()`.
  - Get the user's bet by calling `getUserBet()`.
  - Each round, determine if the user...
    - won the round: award points from the user;
    - lost the round: subtract points from the user;
    - tied the round: no points awarded or subtracted.
    - In each case, display a message that is green (for win), red (for loss), or blue (for tie).
  - Only track the score of the user; do not track the score of the dealer.

- When the user wins or loses the game display an appropriate message and end the game.
  - Display in bold red if the user lost, display in bold green if the user won.
  - 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. For example, the user only tries to enter numbers 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).

Extra colours:
You must colour some additional output to the user to help make the game interface clearer. You must colour at least three things beyond the coloured messages listed above. For example, colour the dice rolls and point total each round.

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 a ZIP file of the items listed below to the CourSys: [https://courses.cs.sfu.ca/](https://courses.cs.sfu.ca/)

1. **co2.cpp**
2. **beattheroll.cpp**
3. Any needed .h files (such as for ANSI codes).

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.