

# EECE 1080C / Programming for ECE

Summer 2019

## Laboratory 5: Game of Chance

**Plagiarism will not be tolerated:**

**all students who share files will receive a 100% penalty on this assignment**

### **Topics covered:**

- Input validation.
- Decision statements.
- Loops and functions.

### **Objective:**

- To gain experience with functions and modular programming.

### **Collaboration:**

- As with most laboratory assignments in this course this laboratory assignment is to be performed by an individual student. You can help each other learn by reviewing assignment materials, describing to each other how you are approaching the problem, and helping each other with syntax errors. You can get help from teaching assistants and instructors. Having slightly similar code for some assignments is expected but most assignments have multiple different solutions. Your code is expected to be different.
- Please document any help you receive from teaching assistants or instructors. Just add the names to the top of your source file.
- Having someone else code for you, sharing code with other students, or copy-pasting code from the internet or previous terms, **is cheating**.

### **Highlights:**

- Please consider the following:
  - Review functions and loops.
  - Read each specification carefully.
  - **Format your program source code properly.**
- To receive full credit for this laboratory please sign the attendance sheet.
- Please access the laboratory assignment via the canopy/blackboard link. The description for the assignment is contained within this document.
- **Submit on Blackboard using the assignment dropdown.**

- The following assignment heavily emphasizes on input validation and decision statements. Be sure to comment your code in order to troubleshoot faster in case it is needed. It is also a very good programming practice.
- Add a main comment section at the beginning of each program you submit, specifying your name in Pinyin, the date it was last updated and a brief description of what it does.

### **Rubric: 100 points**

- Part A = 25
- Part B = Part A + 25
- Part C = Part B + 25
- Part D = Part C + 25

### **Specifications:**

- Your program will run a simulation of a game of chance based on a simple dice game. The code developed in this lab should all be a part of a single program. Each task's code will build upon previous tasks to create the final program at the end of Task D. **You only have one program to submit at the end of this lab.**
- **Game rules:**
  - You pick a number from 1 to 6, and call this number your point. You then roll a die 3 times. If there is any match with your point, then you win 10 RMB; otherwise you pay the house 10 RMB.
- **Program development:**
  - The use of functions is entirely up to the programmer and it is not fully required. The program may contain only the **main()** function.
  - Some basic suggestions for other functions might include the following:
    - Functions to perform input.
    - Functions to display statistics.
    - Functions to handle the complete logic of the program.
  - If you use functions, they will need to make sense in the context of the program.
  - Notes:
    - Avoid using non-constant global variables.
    - **Comment the code appropriately.**
    - All program inputs should be verified for proper range of values. Have other users test your inputs.
    - All program outputs should be properly formatted and readable. Have other users look at your outputs.

## Tasks:

---

### A. Task A

- Design a program to simulate one play of this game. The user should be asked to enter his name. Consider using the random function to obtain values between 1-6. You should also prompt for the point and roll the dice up to three times to see if you win/lose.
- **Hint:** Remember that to utilize random numbers you need to:
  - #include specific libraries.
  - Seed the random function.
  - Call the random function and scale it down to the required numbers.

---

### B. Task B

- If your game works correctly you should then design a loop control structure that keeps repeating the game, and tracks the number of wins and losses. The loop should continue until the user enters in an appropriate sentinel value, such as the character '0' for exit. Verify that the program runs properly, and as expected, before continuing.

---

### C. Task C

- Modify the program so that the user is able to maintain a bank account (which is initialized with 100 RMB) and wager on each play. The program should prompt the user to enter a wager. This wager replaces the simple case of winning/losing 10 RMB from before. Do an error check (using a while loop), so that the wager entered is no more than the current bank balance. If the wager is valid, then inform the user that they will be playing a single game. If the wager is invalid, print an error message and repeat the process.

---

### D. Task D

- Winning the game will increase the bank balance by the wager, and likewise a loss will decrease the balance by same. After the game is played, print the current balance in an appropriate format. If the balance is greater than zero, prompt the user to play again. Continue another round of the game until the user responds that the user finishes playing (by wagering 0), or if the user's balance is zero.
- After the game is over you should print out the number of games won or lost, and the percentage of games won. You may want to create a function to output these statistics.

- **Example Program Run 1:**

Please enter your name: Daniel

Account balance 100 RMB

Enter wager (0 to exit): 50

Enter your point value (1 - 6): 1

Roll #1 is 1

\*\* You win! \*\*

Account balance 150 RMB

Enter wager (0 to exit): 50

Enter your point value (1 - 6): 1

Roll #1 is 5

Roll #2 is 4

Roll #3 is 5

\*\* You Lose \*\*

Account balance 100 RMB

Enter wager (0 to exit): 50

Enter your point value (1 - 6): 1

Roll #1 is 4

Roll #2 is 2

Roll #3 is 6

\*\* You Lose \*\*

Account balance 50 RMB

Enter wager (0 to exit): 50

Enter your point value (1 - 6): 1

Roll #1 is 5

Roll #2 is 1

\*\* You win! \*\*

Account balance 100 RMB

Enter wager (0 to exit): 100

Enter your point value (1 - 6): 5

Roll #1 is 4

Roll #2 is 2

Roll #3 is 6

\*\* You Lose \*\*

Player name: Daniel

Game over, you are out of money!

Your final account balance is 0 RMB

You won 1 time out of 4 for a winning percentage of 25%

- **Example Program Run 2:**

Please enter your name: Daniel

- Account balance 100 RMB

Enter wager (0 to exit): 75

Enter your point value (1 - 6): 4

Roll #1 is 2

Roll #2 is 4

\*\* You win! \*\*

Account balance 175 RMB

Enter wager (0 to exit): 0

Player name: Daniel

Your final account balance is 175 RMB

You won 1 time out of 1 for a winning percentage of 100%

- **Example Program Run 3:**

Please enter your name: Daniel

Account balance 100 RMB

Enter wager (0 to exit): -4

Error: You must wager between \$1 and \$100 (type 0 to exit): 101

Error: You must wager between \$1 and \$100 (type 0 to exit): 45

Enter your point value (1 - 6): -1

Error: Enter your point value, must be between 1 and 6: 1723

Error: Enter your point value, must be between 1 and 6: 234

Error: Enter your point value, must be between 1 and 6: 7

Error: Enter your point value, must be between 1 and 6: 6

Roll #1 is 3

Roll #2 is 1

Roll #3 is 3

\*\* You Lose \*\*

Account balance 55 RMB

Enter wager (0 to exit): 0

Player name: Daniel

Your final account balance is 55 RMB

You won 0 time out of 1 for a winning percentage of 0%