**Every Boilermaker Engineering Code – Entry Level Programming**

**Week 5 – Programming Exercises**

1. **(15 points, Math Quiz)** Write a program that gives simple math quizzes. The program should display two random integers (0-999) that are to be added, such as:

**247**

**+ 129**

The program should allow the student to enter the answer. If the answer is correct, a message of congratulations should be displayed. If the answer if incorrect, a message showing the correct answer should be displayed.

**Test your program 3 times and take the screenshot for all the results. Here is a sample of test. You should follow the format with random generated numbers from your own code, and because the numbers are random, your program may display different outputs with the simple (Only the red numbers should be typed in the interactive mode). Please make sure the two integers are properly aligned.**

|  |
| --- |
| **367**  **+ 941**  **Enter sum of numbers: 1308**  **Correct answer – Good Work!** |
| **48**  **+ 863**  **Enter sum of numbers: 948**  **Incorrect... The correct answer is: 911** |

1. **(15 points, Guessing Game)** Write a program that generates a random integer in the range of 1 through 10, and asks the users to guess that the number is. If the user’s guess is higher than the random number, the program should display “Too high, try again.” If the user’s guess is lower than the random number, the program should display “Too low, try again.” If the user guesses the number, the application should congratulate the user, then the program should generate a new random number so the game can start over. Until the users enter 0, quit the program.

**Test your program 3 times and take the screenshot for all the results. Here is a sample of test. You should follow the format with random generated numbers from your own code,** **and because the numbers are random, your program may display different outputs with the sample (Only the red numbers should be typed in the interactive mode).**

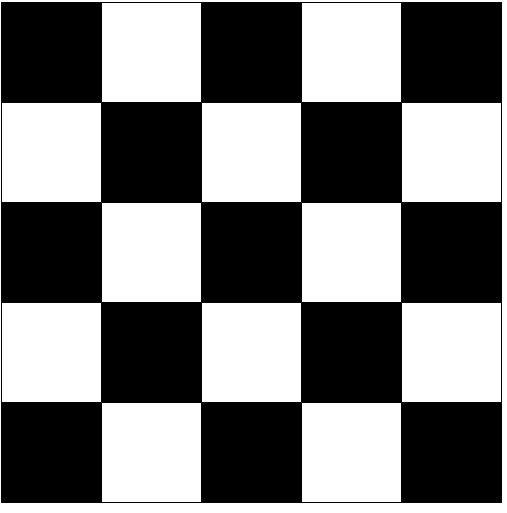
|  |
| --- |
| **Enter a number between 1 and 10, or 0 to quit: 13**  **You guess is out of range, try again.**  **Enter a number between 1 and 10, or 0 to quit: 8**  **Too high, try again.**  **Enter a number between 1 and 10, or 0 to quit: -6**  **You guess is out of range, try again.**  **Enter a number between 1 and 10, or 0 to quit: 6**  **Too high, try again.**  **Enter a number between 1 and 10, or 0 to quit: 3**  **Too high, try again.**  **Enter a number between 1 and 10, or 0 to quit: 1**  **Too low, try again.**  **Enter a number between 1 and 10, or 0 to quit: 2**  **Congratulations! You guessed the right number!**  **A new random integer is generated.**  **Enter a number between 1 and 10, or 0 to quit: 8**  **Too high, try again.**  **Enter a number between 1 and 10, or 0 to quit: 0**  **Thanks for playing!** |

1. **(15 points, Rock Paper Scissors Game)** Write a program that lets user play the game of Rock, Paper, Scissors against the computer. The program should work as follows:
   1. When the program begins, a random number in the range of 1 through 3 is generated. If the number is 1, then the computer has chosen rock. If the number is 2, then the computer has chosen paper. If the number is 3, then the computer has chosen scissors. (Don’t display the computer’s choice yet.)
   2. The user enters his or her choice of “rock”, “paper”, or “scissors” at the keyboard.
   3. The computer’s choice is displayed.
   4. A winner is selected acceding to the following rules:
      * If one player chooses rock and other player chooses scissors, then rock wins. (Rock smashes scissors.)
      * If one player chooses scissors and the other player choose paper, then scissors wins. (scissors cut paper.)
      * If one player chooses paper and the other play chooses rock, then paper wins. (Paper wraps rock.)
      * If both players make the same choice, the game must be played again to determine the winner.

**Test your program 3 times and take the screenshot for all the results. Here is a sample of test. You should follow the format with random generated numbers from your own code, and because the numbers are random, your program may display different outputs with the simple. (Only the red numbers should be typed in the interactive mode).**

|  |
| --- |
| **Enter 1 for rock, 2 for paper, 3 for scissors: 0**  **Computer chose paper**  **You chose something went wrong**  **You made an invalid choice. No winner** |
| **Enter 1 for rock, 2 for paper, 3 for scissors: 3**  **Computer chose scissors**  **You chose scissors**  **You made the same choice as the computer. Starting over**  **Enter 1 for rock, 2 for paper, 3 for scissors: 2**  **Computer chose paper**  **You chose paper**  **You made the same choice as the computer. Starting over**  **Enter 1 for rock, 2 for paper, 3 for scissors: 1**  **Computer chose paper**  **You chose rock**  **The computer wins the game.** |

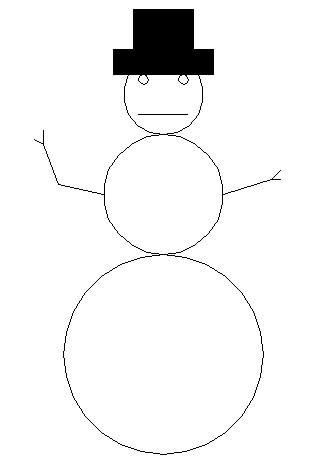
1. **(15 points, Turtle Graphics: Checkerboard)** Write a turtle graphics program to draw a checkboard with user input number of squares, similar to the following figure. In addition to a main function, the program should also meet the following requirements:
   * Include a **drawSquare** function: This function takes four parameters, the **x, y** coordinates of the start location of the square, the **width** of the square, and the **color** of the square.
   * Uses the **drawSquare** function, along with a **loop** (or loops) to the checkboard pattern.



**Use the following numbers to test (Only type in the red numbers in interactive mode):**

|  |  |
| --- | --- |
| **Sample Input** | **Expected Output** |
| **Enter the number of squares in one side: 9** |  |

1. **(20 points, Turtle Graphics: Modular snowman)** Write a program that uses turtle graphics to display a snowman, similar to the following figure (on next page). In addition to a main function, the program should also have the following functions:
   * **drawBase**. This function should draw the base of the snowman, which is the large snowball at the bottom.
   * **drawMidSection**. This function should draw the middle snowball.
   * **drawArms**. This function should draw the snowman’s arms.
   * **drawHead**. This function should draw the snowman’s head, with eyes, mouth, and other facial features you desire.
   * **drawHat**. This function should draw the snowman’s hat.



1. **(Bonus question: 20 points)** Write a turtle graphics program that draws a city skyline similar to the one shown in the following figure. The program’s overall task is to draw an outline of some city buildings against a night sky. Modularize the program by writing the functions that perform the following tasks:
   * Draw the outline of the buildings.
   * Draw some windows on the buildings.
   * Use randomly placed dots as the stars (Make sure the stars appear on the sky, not on the buildings).

