Checkers Game:

Outline:

* Init the global variable that will contains the essential elements of a checker board
  + Board 8x8
  + Newboard 8x8
  + blackPiece
  + redPiece
  + Empty
  + Invalid Slot
  + kingBlack
  + kingRed
* init a class checkerboard that contains instructions to display and init the board
  + create a void function to init board
    - Board 8x8
    - Black and white (invalid vs empty)
    - First 3 rows and 8 columns are the red team side
    - 3-4 rows and 8 columns are the empty slot
    - 5-8 rows and 8 columns are the black team side
  + Create a void function to print the standard board
  + Create a void function to print the new board that contains new values to display
* Create a Boolean function for checking gameover condition
  + to check; use a for loop to loop all the cell
    - if board does not contain blackPiece , redPiece, kingRed, or kingBlack
      * then game is over
    - if board does contain blackPiece , redPiece, kingRed, or kingBlack
      * then game is continuing
* create a void function for red team move
  + init the needed variables
    - integer pieceX
    - integer pieceY
    - integer moveX
    - integer moveY
    - integer decision
    - integer playerOne
  + prompt the user that player one is red team and is beigning
  + ask the user for x and y cords
  + ask the user for x and y cords for new location
  + compare the cords to fit the conditions in the if statement
    - if the piece is red piece
      * if capture is possible and promotion to king is possible
        + set the midpoint enemy to empty
        + set the new location piece red becomes king red
        + set the old location
        + if double jump is possible

ask user to pick yes or no

if yes, loop back

if no, break the loop

* + - * if regular capture
        + set the old location piece empty
        + set the new location piece to red piece
        + set the enemy location to empty by midpoint
        + give playerOne score + 1
        + ask user to pick yes or no

if yes, loop back

if no, break the loop

* + - else the piece is king red
      * set the old location piece empty
      * set the new location piece to red piece
      * set the enemy location to empty by midpoint
      * give playerOne score + 1
      * ask user to pick yes or no
        + if yes, loop back
        + if no, break the loop
  + regular movement condition
    - king creation
      * if the new board spot is empty and that the y cord is 7
        + piece location set empty
        + piece new location set king red
        + break loop
    - regular move
      * if the piece is red piece
        + piece location set empty
        + piece new location set red
        + break loop
    - king move
      * if the piece is king red piece
        + piece location set empty
        + piece new location set king red
        + break loop
  + else cords are incorrect
    - loop back to ask again for the cords until return true
* create a void function for black team move
  + init the needed variables
    - integer pieceX
    - integer pieceY
    - integer moveX
    - integer moveY
    - integer decision
    - integer playerTwo
  + prompt the user that player one is black team and is beginning
  + ask the user for x and y cords
  + ask the user for x and y cords for new location
  + compare the cords to fit the conditions in the if statement
    - if the piece is black piece
      * if capture is possible and promotion to king is possible
        + set the midpoint enemy to empty
        + set the new location piece red becomes king black
        + set the old location
        + if double jump is possible

ask user to pick yes or no

if yes, loop back

if no, break the loop

* + - * if regular capture
        + set the old location piece empty
        + set the new location piece to black piece
        + set the enemy location to empty by midpoint
        + give playertwo score + 1
        + ask user to pick yes or no

if yes, loop back

if no, break the loop

* + - else the piece is king black
      * set the old location piece empty
      * set the new location piece to black piece
      * set the enemy location to empty by midpoint
      * give player two score + 1
      * ask user to pick yes or no
        + if yes, loop back
        + if no, break the loop
  + regular movement condition
    - king creation
      * if the new board spot is empty and that the y cord is 0
        + piece location set empty
        + piece new location set king black
        + break loop
    - regular move
      * if the piece is black piece
        + piece location set empty
        + piece new location set black
        + break loop
    - king move
      * if the piece is king black piece
        + piece location set empty
        + piece new location set king red
        + break loop
  + else cords are incorrect
    - loop back to ask again for the cords until return true
* Create the main function where it executes
  + Create variables that will be within the main function
    - Integer Choice – player team choice
    - Integer Begin – start game decision
    - Integer Player one – start score as 0
    - Integer player two – start score as 0
  + Initialize the file stream with created object checkerboard
  + Initialize the class checkerboard with created object checkerboard
  + Display the rules and layout of the menu
  + Ask user if want to begin or not
    - Store it in the begin variable
      * If yes
        + Init checkerboard
        + Print checkerboard
        + Create a for loop to loop through the 8x8 array to set the elements to the newBoard array 8x8

//this will be user input

* + - * + Ask user to pick team red or team black

If team red

Prompt the user that player one is red

Prompt the user that plater two is black

Else team black

Prompt the user that player one is red

Prompt the user that player two is black

* + - * + Create a do while loop which continues to loop as long as it is not gameOver

Init the variables inside the while loop that will be used

Integer piecex

Integer piecey

Integer moveX

Integer moveY

Integer decision – to jump again

Integer playerOne – score

Integer playerTwo – score

Create a for loop that will loop about 25 times until condition of gameover is true

Call the void function redMove

Print new board array

Call the void function blackMove

Print new board array

Display the score of players one and player two

If condition of gameover is true

Open text file for storing the score and display of file

Close file

* + - * Else no
        + Exit the program

Logic:

* Include the necessary library
* Define variables that are needed for the checkerboard
  + Board 8x8
  + Newboard 8x8
  + blackPiece
  + redPiece
  + Empty
  + Invalid Slot
  + kingBlack
  + kingRed
* Initialize checkerboard [class] //can also use structure instead of class since there is no need for private variables
  + Public variables or instructions
    - Initialize board //first layout of board
      * Board 8x8
      * Black and white (invalid vs empty)
      * First 3 rows and 8 columns are the red team side
      * 3-4 rows and 8 columns are the empty slot
      * 5-8 rows and 8 columns are the black team side
    - Print board //display the board with unique design
      * Print board is the standard board
    - Print new board //display the new board that contains the standard board
      * New board is the getBoard, input from user
* Initialize a condition in which the game is over (gameOver)
  + (True or false) winner is currently false
    - While it is false
      * Loop through the 2d array to check for
        + blackpiece/kingBlack or redPiece/kingRed

if true, turn winner to true which ends the condition

if false; turn winner to false to continue the iteration or condition

* define function redMove (to check move and conditions)
  + init variables
    - pieceX – original x cord placement
    - pieceY – original y cord placement
    - moveX – new set x cord placement
    - moveY – new set y cord placement
    - decision – allow user to jump again for decision
    - playerOne – score of player one if capture a piece
  + Call/define the class checkerboard with an object checkerboard to initiate it’s contents
  + Prompt the user that it is red team’s first and that they have to decide which piece they want to move
  + Create a while loop in case if user made a mistake inputing
    - Ask user for the piece x cord
      * Store the value in pieceX
    - Ask the user for the piece y cord
      * Store the value in pieceY
    - Ask the user for the new location
      * Store in the piece for moveX
    - Ask the user for the new location
      * Store in the piece for moveY
  + Create a condition (if loop) that holds conditions in which it will return true

(If destination of piece is empty, and [(the destination distance is up one and right or left one) or (is a king piece) or (the destination distance is down 2 and right 2 or left 2)]

* + - then split those conditions up to then execute even further (capture condition)
      * create a conditions in which it checks for jump by using the midpoint equation
      * if that is true and the piece is red piece
        + create a if statement which ask if the location is empty and that the new y cord is = 7 //this helps fix take and promotion

set the enemy empty

set the old location empty

set the new location as king red

add a point to playerOne

ask user if they want to jump again if possible

if yes

loop bac kto the while loop and continue from there

if no

break the loop

* + - * + create an else loop statement for //regular capture

set the original piece empty

set the new location piece empty

set location of the midpoint empty

add points to player one

ask user if they want to jump again if possible

if yes

loop back to the while loop and continue from there

if no

break the loop

* + - * if that is true and the piece is king red
        + set the piece location empty
        + set the new location king red
        + set the location at midpoint empty
        + add points to playerOne
        + ask the user if they want to jump again if possible

if yes

loop back to the while loop and continue from there

if no

break the loop

* + - (move condition)
      * King creation
        + If spot is empty and the y cord is 7 (last index)

Set piece location to empty

Set new location to king red

Break the loop

* + - * Regular move
        + If the piece is a red piece

Set the location to empty

Set the new location to red piece

* + - * King move
        + If the piece is a king red piece

Set the location to empty

Set the new location to king red

* + Create an else loop that loops again to ask user again until the condition for the if loop is true
* define function blackMove (to check move and conditions)
  + init variables
    - pieceX – original x cord placement
    - pieceY – original y cord placement
    - moveX – new set x cord placement
    - moveY – new set y cord placement
    - decision – allow user to jump again for decision
    - playerOne – score of player one if capture a piece
  + Call/define the class checkerboard with an object checkerboard to initiate it’s contents
  + Prompt the user that it is red team’s first and that they have to decide which piece they want to move
  + Create a while loop in case if user made a mistake inputting
    - Ask user for the piece x cord
      * Store the value in pieceX
    - Ask the user for the piece y cord
      * Store the value in pieceY
    - Ask the user for the new location
      * Store in the piece for moveX
    - Ask the user for the new location
      * Store in the piece for moveY
  + Create a condition (if loop) that holds conditions in which it will return true

(If destination of piece is empty, and [(the destination distance is down one and right or left one) or (is a king piece) or (the destination distance is up 2 and right 2 or left 2)]

* + - then split those conditions up to then execute even further (capture condition)
      * create a conditions in which it checks for jump by using the midpoint equation
      * if that is true and the piece is black piece
        + create a if statement which ask if the location is empty and that the new y cord is = 0 //this helps fix take and promotion

set the enemy empty

set the old location empty

set the new location as king black

add a point to playerTwo

ask user if they want to jump again if possible

if yes

loop back to the while loop and continue from there

if no

break the loop

* + - * + create an else loop statement for //regular capture

set the original piece empty

set the new location piece empty

set location of the midpoint empty

add points to player two

ask user if they want to jump again if possible

if yes

loop back to the while loop and continue from there

if no

break the loop

* + - * if that is true and the piece is king black
        + set the piece location empty
        + set the new location king black
        + set the location at midpoint empty
        + add points to playertwo
        + ask the user if they want to jump again if possible

if yes

loop back to the while loop and continue from there

if no

break the loop

* + - (move condition)
      * King creation
        + If spot is empty and the y cord is 0 (last index)

Set piece location to empty

Set new location to king black

Break the loop

* + - * Regular move
        + If the piece is a black piece

Set the location to empty

Set the new location to black piece

* + - * King move
        + If the piece is a king black piece

Set the location to empty

Set the new location to king black

* + Create an else loop that loops again to ask user again until the condition for the if loop is true
* Create the main function where it executes
  + Create variables that will be within the main function
    - Integer Choice – player team choice
    - Integer Begin – start game decision
    - Integer Player one – start score as 0
    - Integer player two – start score as 0
  + Initialize the file stream with created object checkerboard
  + Initialize the class checkerboard with created object checkerboard
  + Display the rules and layout of the menu
  + Ask user if want to begin or not
    - Store it in the begin variable
      * If yes
        + Init checkerboard
        + Print checkerboard
        + Create a for loop to loop through the 8x8 array to set the elements to the newBoard array 8x8

//this will be user input

* + - * + Ask user to pick team red or team black

If team red

Prompt the user that player one is red

Prompt the user that plater two is black

Else team black

Prompt the user that player one is red

Prompt the user that player two is black

* + - * + Create a do while loop which continues to loop as long as it is not gameOver

Init the variables inside the while loop that will be used

Integer piecex

Integer piecey

Integer moveX

Integer moveY

Integer decision – to jump again

Integer playerOne – score

Integer playerTwo – score

Create a for loop that will loop about 25 times until condition of gameover is true

Call the void function redMove

Print new board array

Call the void function blackMove

Print new board array

Display the score of player one and player two

If condition of gameover is true

Open text file for storing the score and display of file

Close file

* + - * Else no
        + Exit the program