Project 2

<Blackjack>

CIS 5 -41366

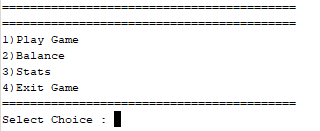
Michael Guerrero

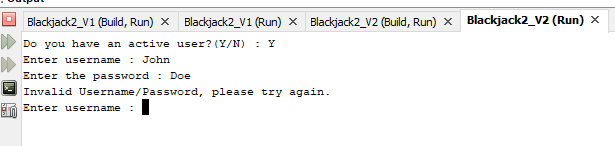
02/14/2021

**Program Summary**

This is my attempt at Blackjack with only using knowledge from the entire class The user starts off at the main menu where they can check their current balance, win stats, and start new games. There is also a login system that will remember the players username and password.

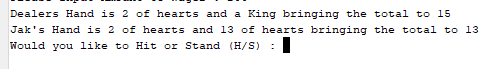
***Example:***





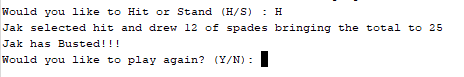
Once assigned their hand the user can determine whether or not they would like to hit or stand.

***Example:***



Which will then display the correct win condition depending on the results.

***Example:***



**Project Size**

This project is about ***409 lines of code***, with around ***24 variables***. The number of lines were able to compress quite a bit including functions and arrays into the program. Though the number of variables seemed to increase.

**Project Shortcomings**

During the creation and updating of the project from the first, I was having difficulty figuring out how I could include 2D Arrays or Vectors and in the end wasn’t able to include that or a lot more. In hindsight I probably should have scrapped most of the program and started back again with the bare bones but trying to force everything to work ended up costing me half of the checklist.

**Project Results**

The final product is a basic working of Blackjack, unfortunately I was not able to add in the ability to double down or split but the core is there. In total this project is not the best but I had a lot of fun learning and implementing stuff we learned in class into a complete program.I gained a lot of insight on how version controls of programs really matter, as during my creation I ran into huge bugs where I would’ve had to reset entirely if not for having different checkpoints.

**Pseudo-Code**

*Initialize Variables*

*Login to User Account/ Create New Account*

*Display Menu*

*Read in users input*

*Ask for wager*

*Randomize Dealer and Players hands as if cards were shuffled*

*Assign Ace,Jack,King and Queen to correct number values*

*If Player > 21 or equal to 21 set bust to true*

*If dealer > 21 or equal to 21 set bust to true*

*If player and dealer are both less than 21 give player option to hit or stand*

*If hit assign player new card and add to total*

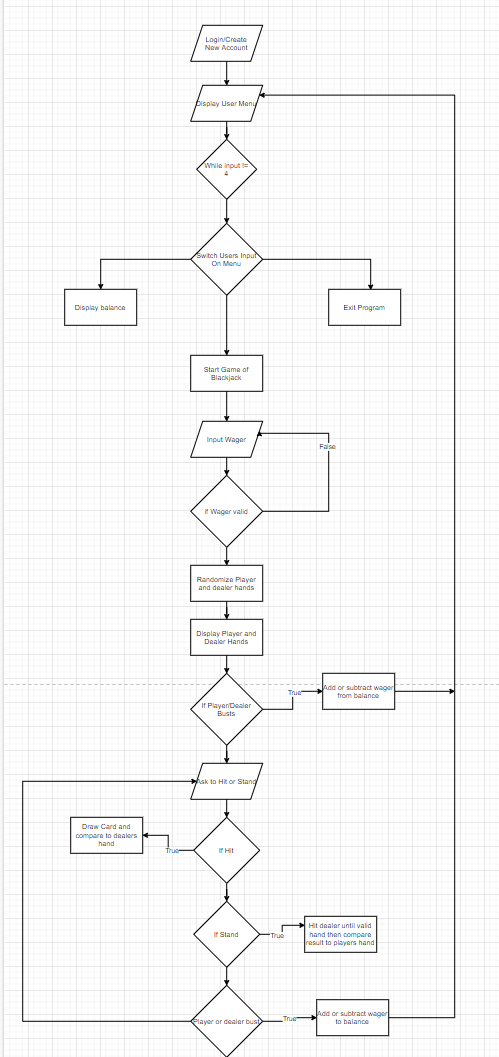
*If stand keep players balance and have dealer pull card until valid hand*

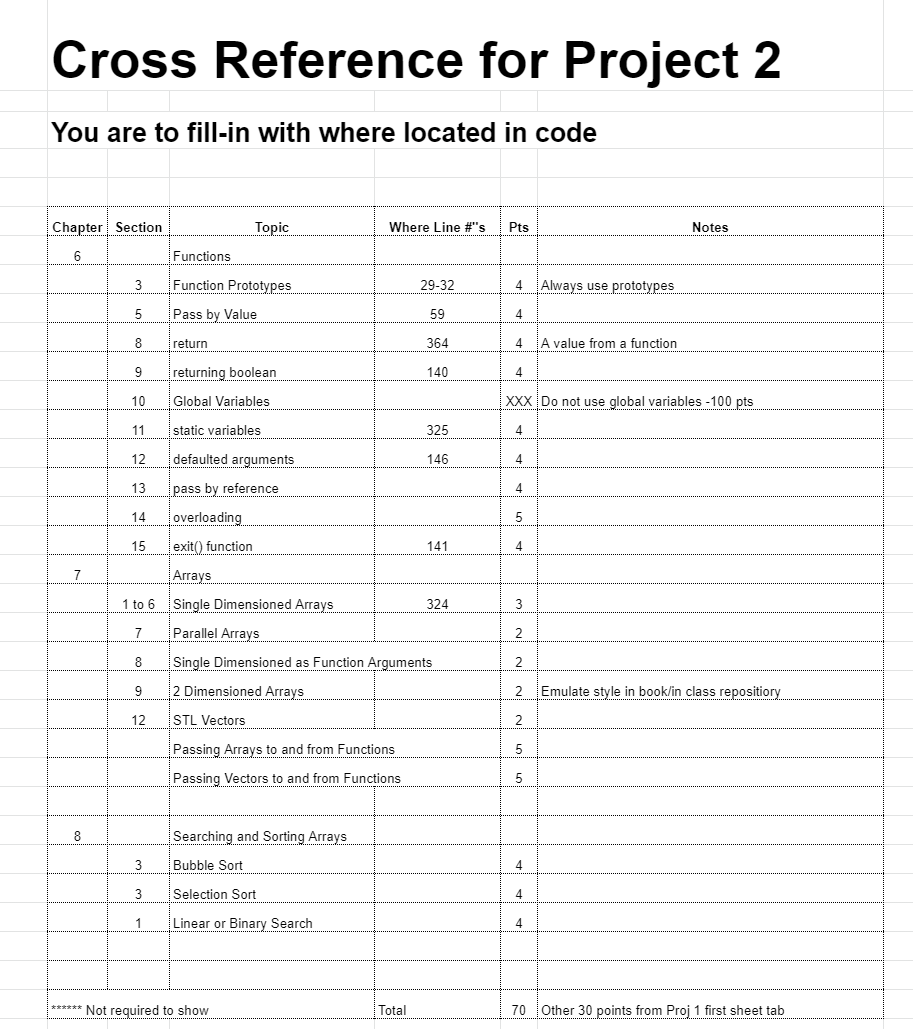
*Calculate the results*

*Add or subtract players wager from balance*

*Return player to Menu*

*Else end program*

**FlowChart** (Hard to see but included in project file)

**Project Check Off List**