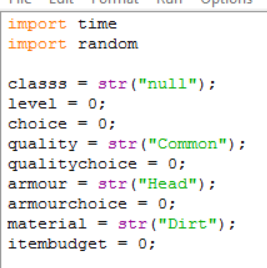
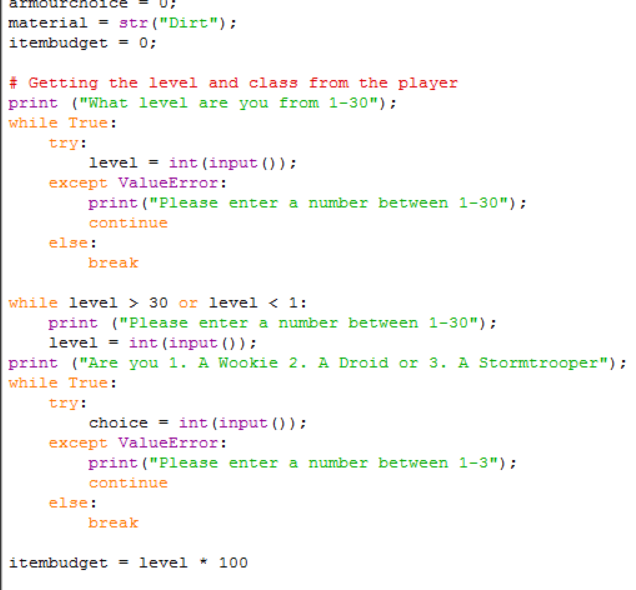
* First all the base variables are set up and time and random functions are imported into the script, the basic variables are the class that will be chosen, level that will be chosen, choice to keep track of the users choice, quality which will be chosen, quality choice to keep track of numerical choice, armour which will be chosen and armourchoice to keep track of the numerical choice for armour, material which will be generated and the overall item budget



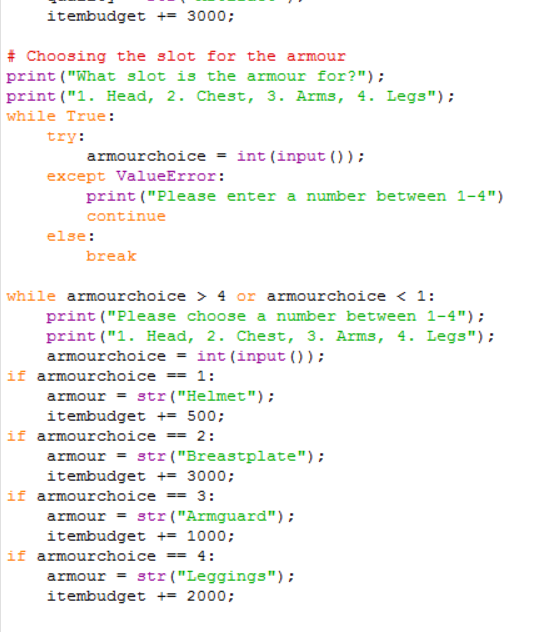
* Then we make the user choose the level from a number between one to thirty, with a while True break code to make sure that a numerical statement is presented and a while level is below 30 and above 1 statement to make sure that an invalid number has not been entered and setting the itembudget to be 100 \* the level chosen



* Also included in that picture is the numerical choice of either 1, being a wookie 2, being a droid or 3, being a stormtrooper with the same while true and while statements to make sure a valid input is applied.
* Next is the numerical choice for quality which takes your numerical choice and change it into a string choice as well as assigning appropriate additions to the item budget (based on testing).



* Next is the numerical choice for the slot of the armour, also converted into a string choice, with the same while true and while statements to make sure of a valid input, item budgets are also assigned due to importance of the choice (for example breastplate is strongest so it gets an item budget of 3000)



* The rest of the code is for generating random values for example effects and attributes are generated from a random choice of 0 to 1 and all decimals inbetween and if they are between two values a certain attribute or status effect is chosen then removed from the list, assuring that no two effects or attributes can be chosen twice.
* Then the item budget is divided by 4 and each quarter is put into its own value assigned itembudget1-4 and used to generate the values of the armour piece as well as status effects and attributes, the four quarters are only used once making sure that the whole item budget is expended in the creation of the armour
* The program then prints all the values that its been storing giving you all the choices you have made and all the generated values to go with them ending with a “press enter to continue” to end the program, the program requires a restart to run again

**Things I would change in a second attempt**

* I would make the whole program in a function loop so the program can be run again without restarting it
* I would implement the itembudget function a lot smoother, misunderstanding it the first time is the primary reason for its sloppy and forced in approach
* Id make possibly buttons to click for choices rather than numerical choices lowering the margin even more for user error
* Lastly possibly give it a UI to look nice instead of just a console python program.