**3. Iteration (Unconditional loops)**

**Task 4:**

**Plan:**

1. Use enters their initial loan, years, and interest percentage. All these values are stored in separate variables.
2. each variable is checked to see if it contains only digits through the use of validation. If it is not an error message is displayed. else if it does contain only digits the while loop ends.
3. Each variable is then cast to an integer, and fed through a for loop. The number of iterations the loop does, depends on the number of years the user enters.
4. After every iteration the for loop does, the interest is calculated and added to the loan from the previous year to form a total. That total then becomes the new loan for the next year.

**Pseudocode:**

loan 🡸 years 🡸 interest 🡸 str()

while str(loan+years+intrest).isdidigt() != True:

loan 🡸 input(“Enter loan: ”)

years 🡸 input(“Enter Years: ”)

interest 🡸 input(“Enter Interest: ”)

if str(loan+years+intrest).isdigit() !=True:

print(“Incorrect Input!”)

loan, years, interest 🡸 int(loan), int(years), int(interest)

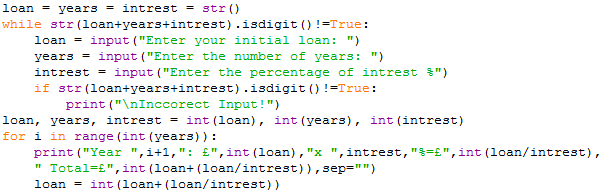
for I in range(years):

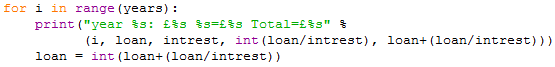
print(“Year”,I,”£”,loan,interest,”£”,loan/interest,”total £”,loan+(loan/interest))

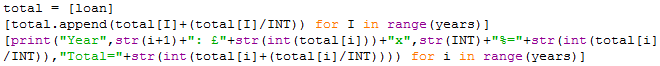
loan 🡸 loan+(loan/interest)

**Variables table:**

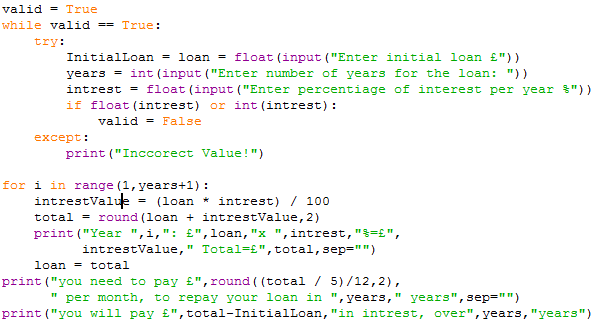
|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data Type** | **Comment** |
| Loan | Integer | Stores inputted loan value. |
| Years | Integer | Store inputted years value. |
| Interest | Integer | Stores inputted interest. |

**Screenshot evidence:**

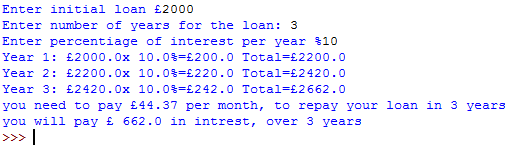
In the screenshot above 3 variables are defined, loan, years, and interest as strings. A while loop then runs, validating the users input as they enter a value for each of the 3 variables. The 3 variables are then checked against a .isdigit() condition, if any of the variable don’t contain only digits then the string “Incorrect Input!” is output to the user, and they are asked to enter the values again. Once the user has entered a valid input for all 3 variables, there data types are cast to integers, allowing for calculation of the interest, and total. A for loop then runs for the number of iterations specified by the “years” variable. Duran each loop, a print statement is used to output information to the user in the following format, "year %s: £%s %s=£%s Total=£%s" % (i, loan, intrest, int(loan/intrest), loan+(loan/intrest)). The total is then assigned in the variable “loan”, allowing for next year’s calculations. As show in the screenshot bellow.

But this is not the only way doing it. The screenshot bellow shows the use of list comprehension. This allows for the for loop and print statements to be potentially on one line. but I have separated them over 3 lines so they fit in the screenshot bellow.

Below is a screenshot of my first attempt at this task. It is not as efficient and cleverly coded as the screenshot above, but does show a clear and easier to follow version of the code. This is because all the calculations are stored in different variables, compared to the code above where everything is built into a print statement.

This code also accurately calculates the total amount of interest you will be paying. And even tells you how much you will need to pay per month to reach your target of No. years.

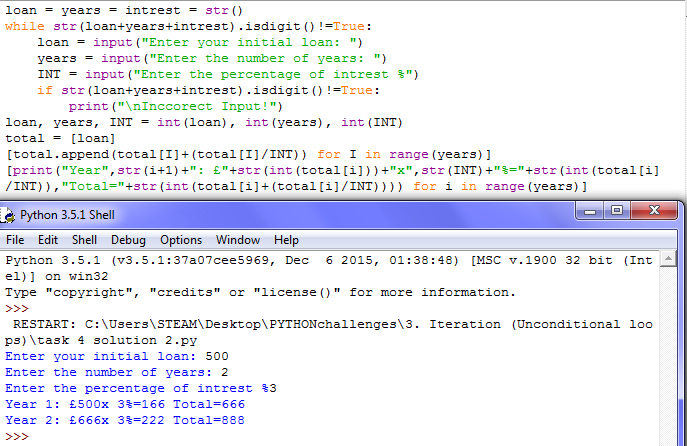
**Sample run (Output):**

Here is the out from the code shown above: - task 4 solution 1

As you can see the user has entered a loan of £2000, which they will pay back over 3 years, with a 10% interest rate. The program has calculated the interest, then added the loan and interest together to form the total. That total then becomes the new loan for the next year. Once all the years have been calculated, the program outputs the amount the user will need to pay per month. for example, in the screenshot above, the user will need to pay £44.37 per month. The program then also prints the total interest, e.g. £662.0.

Task 4 solution 2:

The screenshot bellow shows the coded solution, and its output in the python idle. The user has entered the interest value 500, which they will pay back over 2 years, with a 3% interest rate. This gives a total of £888, which results in the user paying a total of £388 in interest.

**Flowcharts:**

for I in range(years)

print(“Incorrect Input!”)

NO

START

loan 🡸 input(“Enter loan: ”)

years 🡸 input(“Enter Years: ”)

interest 🡸 input(“Enter Interest: ”)

Is (loan+intrest+year).isdigit () = True?

loan, years, interest 🡸 int(loan), int(years), int(interest)

YES

print(“Year”,I,”£”,loan,interest,”£”,loan/interest,”total £”,loan+(loan/interest))

loan 🡸 loan+(loan/interest)

LOOP until I = years

END