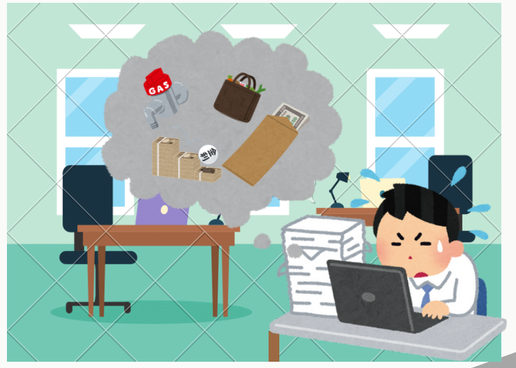
PRELIM CASE STUDY 1- COMPUTATIONAL THINKING W/ PYTHON

NEW VEHICLE AFFORDABILITY PROBLEM

Team 2
Ronel Christian Castillo
Jeremy John Orlina



Problem Context

A recently promoted salaryman has decided that, as a reward for themselves, buy a new vehicle for convenience's sake. But he wants to know whether or not they can afford it with their current monthly salary since they know that a vehicle is an liability, not an investment.

Decomposition

Calculating if the vehicle can be afforded or not, considering if, with its additional expenses involved, can be purchased. Lists monthly payments using a Bottom-up approach. Usage of a Loan Formula is needed.

Pattern Recognition

If Monthly payment < Monthly costs = affordable. If Monthly payment is more than 15% of monthly salary = Cannot Afford.

Abstraction

Relevant Data: Maintenance, insurance fees, Monthly Gas Money, monthly cost of the vehicle overall.

Irrelevant Data: Brand, vehicle type, manufacturer.

Factors to consider/parameters

- SALARY
- PRICE OF NEW VEHICLE
- LOAN TERM
- ANNUAL INTEREST RATE
- MONTHLY INSURANCE
- MONTHLY FUEL COST
- MONTHLY MAINTENANCE COST

Finding the Solution

- USING GIVEN FORMULA TO CHECK FOR COST:

$$P = \frac{PV \times r \times (1+r)^n}{(1+r)^n - 1}$$

- DETERMINING IF MONTHLY COST <= MONTHLY PAYMENT
- COMPARING SALARY TO PAYMENT COST

Inputs

CODE IMPLEMENTATION

```
[40] def getDetails():
         print("Callin' for some Wheels? Lets check if you can afford it!")
         monthlySalary = float(input("Enter your monthly salary: ")) # get inputs for user's monthly salary
         vehiclePrice = float(input("Enter the vehicle price: ")) # get inputs for desired vehicle's price
         loan = int(input("Enter the loan term (in years): ")) # get inputs for the loan term in years
         annualInterestRate = float(input("Enter the annual interest rate (as a percentage): ")) # interest rate to be user later for calculating
         insurance = float(input("Enter the monthly insurance cost: ")) # car necessities that need to be considered
         fuelCost = float(input("Enter the estimated monthly fuel cost: "))
         maintenanceCost = float(input("Enter the estimated monthly maintenance cost: "))

         return (monthlySalary, vehiclePrice, loan, annualInterestRate, insurance, fuelCost, maintenanceCost)
```

Functions and Algorithm

```
[27] def calcPayment(vehiclePrice, loan, annualInterestRate): # function for calculating, we are using a formula given to us by a local dealership
        loanTermMonths = loan * 12
        monthlyInterestRate = (annualInterestRate / 100) / 12


        monthlyPayments = [0] * (loanTermMonths + 1)# list to store monthly payments for calculating

        if monthlyInterestRate == 0:
            monthlyPayments = [vehiclePrice / loanTermMonths] * (loanTermMonths + 1) # condition if interest rate is zero
        else:

            for month in range(1, loanTermMonths + 1):
                monthlyPayments[month] = (vehiclePrice * monthlyInterestRate * (1 + monthlyInterestRate)**month) / ((1 + monthlyInterestRate)**month - 1) # calculate monthly payments

        return monthlyPayments[-1]  # return last monthly payment
```

```
def monthlyCost(monthlyPayment, insurance, fuelCost, maintenanceCost):
    return monthlyPayment + insurance + fuelCost + maintenanceCost
```

```
def canIGetIt(monthlySalary, totalMonthlyCost, percentage=15):
    iWantToKnow = (percentage / 100) * monthlySalary
    return totalMonthlyCost <= iWantToKnow
```

Sample Outputs

```python
def newWheels():
    monthlySalary, vehiclePrice, loan, annualInterestRate, insurance, fuelCost, maintenanceCost = getDetails()

    # Calculate monthly payment using dynamic programming approach
    monthlyPayment = calcPayment(vehiclePrice, loan, annualInterestRate)

    # Calculate total monthly cost
    totalMonthlyCost = monthlyCost(monthlyPayment, insurance, fuelCost, maintenanceCost)

    # Check affordability
    canIGetIt_result = canIGetIt(monthlySalary, totalMonthlyCost)

    # Display result
    if canIGetIt_result:
        print("Nice! You can afford this vehicle!")
    else:
        print("You cannot afford this vehicle based on your current monthly salary.")

newWheels()
```

```
Callin' for some Wheels? Lets check if you can afford it!
Enter your monthly salary: 5000
Enter the vehicle price: 15000
Enter the loan term (in years): 3
Enter the annual interest rate (as a percentage): 4
Enter the monthly insurance cost: 150
Enter the estimated monthly fuel cost: 100
Enter the estimated monthly maintenance cost: 50
Nice! You can afford this vehicle!
```

```python
newWheels()
```

```
Callin' for some Wheels? Lets check if you can afford it!
Enter your monthly salary: 4000
Enter the vehicle price: 15000
Enter the loan term (in years): 2
Enter the annual interest rate (as a percentage): 4
Enter the monthly insurance cost: 220
Enter the estimated monthly fuel cost: 140
Enter the estimated monthly maintenance cost: 220
You cannot afford this vehicle based on your current monthly salary.
```

colab notebook link:https://colab.research.google.com/drive/IBqinmuHfmRG6GzTwpFuhPvFdwA6XsR-q#scrollTo=lLr_FbhYWwfb

CPE 311-CPE22S1 | COMPUTATIONAL THINKING W/ PYTHON

THANK YOU

Presented By : TEAM 2

CASTILLO, RONEL CHRISTIAN

ORLINA, JEREMY JOHN