Design Document Assignment 1

Name: Tom Nyuma
ID: 934044293
CS161 Section 015
Date: 01/09/2021

Problem Statement: We are to write a C++ program that reads multiple inputs from a car salesperson. The input will contain data values that can be used to thus calculate their Gross Income and Taxes.

Understanding the Problem: The three main outputs of the program will be the specific salesperson's Gross Income & Taxes, plus any remaining Income. To get these values, we will need inputs consisting of all the necessary variables needed to calculate their respective output. I am going to assume that the salesperson is only responsible for selling one specific car model.

The input will be as follows:

- Monthly Salary
- # of months worked in a year
- Cost of a car
- Number of cars sold
- # of misconducts observed
- An option of the tax year(s)
- A state of choice

Based on these input values, we can assume the data types for most if not all of the input. I can assume all the input value(s) data types are unsigned because we have no need for non-positive values.

All we need to do is take each of the input values and plug them into the right context of the greater equation in question. The first step would be calculating Gross Income. After that, we can use the remaining inputs and return value (gross income) to calculate our taxes and any remaining income we have for the year.

Pseudocode:

```
/*************************
Program Filename: assignment1.cpp
Author: Tom Nyuma
Date: 01/09/2021
Description: Program calculates the gross income and taxes for a salesperson
Input : (See Above)
Output: Gross Income, Taxes and, remaining income after taxes.
*************************/
```

```
// This function calculates Annual Salary based on input and returns it.
func calculate_annual_salary(monthly_pay,months_worked)
      {
            int annual_salary = monthly_pay * months_worked
            return annual_salary
      }

// This function calculates Profit based on more input given and returns it
at the end of the calculation.
func calculate_profit(cars_sold,cost_car)
      {
            double average_sell_price = rand() % 10 + 5
            profit = cars_sold * (average_sell_price - cost_car)

            return profit
      }

// This function uses the misconducts_observed variable and pow() function to
get the deductions that will be used to calculate gross income.
func calculate_deduction(misconducts_observed)
      {
            if (misconducts_observed == 0)
            {
                  return 0
            } else {
                  deduction = 100 * (pow(2,misconducts_observed - 1));
            }
      return deduction
      }
// This function gets 2% of the profit (in this case) and returns the value
that will be added with our annual salary.
func x_percent_profit(double profit)
      {
            double profit_plus;
            return (profit / 100) * double profit_pluss;
      }
// This is the main function to calculate gross income. It uses the formula
given to us to calculate the gross income and return the value to soon be
used with our taxes.
func calculate_gross_income()
      {
            unsigned double gross_income = annual_salary +
      x_percent_profit(profit) - calculate_deduction(20)
```

```
            return gross_income;
      }

// This function, the main function, will contain the logic for our taxes to
keep the pseudocode messiness to a minimum.
func main()

{
      unsigned short int tax_year_opt;
      cout << "Which tax option are you in ? 1 for 2017 2 for 2018: \n";
      cout << "Which state are you in ? A, B, C";

      cin >> tax_year_opt >> char state_opt ;

      if (tax_year_opt == 1 & state_opt == "A") {
            return 6% of gross_income
      } else if (tax year_opt == 1 & state_opt == "B") {
            if (makes < 2000)
            {
                  int taxes = 0
            }
            else if (makes > 2000 & < 10000)
            {
                  int taxes = 100;
            } else {

                  int taxes = 100
                  return taxes + 10% of gross_income
            }
      } else if (tax_year_opt == 1 & state_opt == "C")
            {
                  if (makes < 3500) {
                        return 5% of gross_income
                  } else if (makes < 9000 & > 3500) {
                        int taxes = 175;
                        return taxes + 7% of gross_income
                  } else if (makes < 125000 & > 9000) {
                        int taxes = 560;
                        return taxes + 9% of gross_income
                  } else {
                        int taxes = 11000;
                        return taxes + double 9.9% of gross_income
            }
```

```
	} else if (tax_year_opt == 2 & state_opt == "A")
	{
		return 8% of gross_income
	} else if (tax_year_opt == 2 & state_opt == "B")
	{
		if (makes < 2500) {
			return 0;
		} else if (makes < 100000 & > 2500)
		{
			int taxes = 115;
		} else {
			int taxes = 115;
			return taxes + 10.5% of gross_income
		}
	}else if (tax_year_opt == 2 & state_opt == "C"){
		if (makes < 3450){
			return 5% of gross income
		} else if (makes > 3450 & < 8700)
		{
			double taxes = 172.5;
			return taxes + 7% of gross_income
		} else if (makes > 8700 & < 125000) {
			int taxes = 540;
			return taxes + 9% of gross_income
		} else {
			int taxes = 11007;
			return taxes + float(9.9%) of gross_income
		}
	}
	float remaining_income =  gross_income - taxes;
	return taxes + x_percent_of_income(), reminaing_income;

}
```

Testing: (Good Input)

Value (aka. Input)	Expected Output / Prompt	Does it meet Expectations?

unsigned int monthly_salary = 5000;	A non-negative integer. Prompt months_worked	Yes
unsigned short int months_worked = 11;	A non-negative integer between 0 and 12 Prompt car_cost	Yes
unsigned int car_cost = 60000;	A non-negative integer that could be larger than 2 bytes. Prompt # of cars sold	Yes
unsigned short int num_cars_sold = 20;	A non-negative value, prompts misconducts_observed	Yes
unsigned short int misconducts_observed = 2;	A non-negative 2-byte value. Prompt tax_year_opt	Yes
unsigned short int tax_year_opt = 1;	A non-negative 1-byte value. (Either 1 or 2) With 1 being 2017 and 2 being 2018. Prompt state_opt	Yes
unsigned char state_opt = A;	A non-negative 1-byte value. (Either A, B, or C) which each have different tax dividends. std::cout << gross_income,tax_need_pay,reaminging_income	Yes

Testing: (Bad/Edge Input)

Bad Input(s)	Expected Output / Prompt	Does it meet Expectations?
unsigned int monthly_salary = -1;	A non-negative integer. Prompts "Error, positive input expected \n";	No
unsigned short int months_worked = 13;	A non-negative integer between 0 and 12 Prompts "Error, months must be in a one	No

	year range (0-12m)\n";	
unsigned int car_cost = -1;	A non-negative integer that can be larger than 2 bytes. Prompts "Error, unsigned integers must be positive\n";	No
unsigned short int num_cars_sold = 70000;	A non-negative value, prompts "Error, short int must be 2 bytes"	No
unsigned short int misconducts_observed = -1;	A non-negative 2-byte value. Output"Error, misconducts_observed must be positive"	No
unsigned short int tax_year_opt = 3;	A non-negative 2-byte value between a range of 1 and 2 inclusive. (Either 1 or 2) With 1 being 2017 and 2 being 2018. Output "Input must be either a 1 - 2017, or 2 - 2018."	No
unsigned char state_opt = D;	A non-negative 1-byte value. (Either A, B, or C) which each have different tax dividends. Output "D is not an option" std::cout << gross_income,tax_need_pay,reaminging_income	No