# 1DV533 STEP 1 Assignment report

## Jesper Malmberg em222vs@student.lnu.se

## Task 3

There where three errors in the code that I could find and I marked them with \*FIXED\* as shown in the screenshot below.

//-----------------------------------------------------------------------

// Object: This program has some syntax errors. Use the compiler to find

// and correct them. Document your changes with comments or in log.

//-----------------------------------------------------------------------

// File: CompileError.cpp

// Summary: This program calculates the weekly pay, given hours per week

// and wages per hour.

// Version: 1.1

// Owner: Tommy L�fqvist

//---------------------------------------------------------------------

// Log: 2005-03-09 Created the file, Mats!

// 2009-08-14 Revised by Tommy.Using manipulators to format output

// instead of IO-flags. Inserted "wait for return code" at the end. // 2012-10-21: Revised by Anne. Converted to English version and VS2012

//-----------------------------------------------------------------------

#include <iostream> // Preprocessor directives

#include <iomanip>

using namespace std;

int main() // \*FIXED\* main should return an int not void.

{

    // Define and initialize variables

    int hoursPerWeek = 35;

    double hourlyWages = 83;

    // Calculate weekly salary

    int weeklyWages = hoursPerWeek \* hourlyWages; // \*FIXED\* weeklyWages should be of datatype int.

    // Display results

    cout << fixed // Floating point format

        << setprecision(2) // 2 decimals

        << showpoint; // Show trailing zero�es

    cout << " Given an hourly wage of " << hourlyWages << " kr" << endl

        << " and the number of hours per week " << hoursPerWeek << endl

        << "the weekly wages will be: " << weeklyWages << " kr" << endl; // \*FIXED\* added missing semi-colon at the end of this expression.

    cout << "\nPress return!";

    cin.get(); // Wait for return

    return 0;

}

## Task 4

I found 4 logical errors and marked them with \*FIXED\* next to where I changed the source code.

//-----------------------------------------------------------------------

// Object: This program has logical errors. Find and correct them.

// There are miscalculations and the program can't be stopped with 'N'.

// Document your changes with comments or in log.

//-----------------------------------------------------------------------

// File: LogicalError.cpp

// Summary: Reads price and number of articles from user. Calculates and prints

// number, tax rate and price with tax rate (swedish "moms").

// Version: 1.1

// Owner: Tommy L�fqvist

//-----------------------------------------------------------------------

// Log: 2005-03-09 Created the file, Mats!

// 2009-08-14 Revised by Tommy. Swedish variable names and output text.

// Number of items is now included.

// Code for the problem is placed in a separate function

// that is iterated in the main function.

// 2012-10-21 Revised by Anne. Converted to English version and VS 2012

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

using namespace std;

// Prototypes

void priceCalculation();

int main()

{

    char answer;

    do

    {

        priceCalculation();

        cout << "One more time? (Y/N): ";

        cin >> answer;

    } while (answer == 'Y' || answer == 'y'); // \*FIXED\* was answer = 'Y', should have two '='.

    return 0;

}

//-----------------------------------------------------------------

// void priceCalculation()

//

// Summary: Reads price and number of articles from user. Calculates and prints

// quantity, vat and price with vat

// Returns: -

//-----------------------------------------------------------------

void priceCalculation()

{

    // Define and initialize constants and variables

    const double RATE = 25;     // tax rate in percent

    //\*FIXED\* changed from int to double to avoid 'rounding'.

    double price = 0;           // price per piece

    int nrOfArticles = 0;   // number of articles

    double rateSEK = 0;         // tax rate in SEK

    double totalPrice = 0;      // price incl. tax rate

    // Read price and number of articles

    cout << "Enter the price excl. the tax rate (swedish moms): ";

    cin >> price;

    cout << "Enter the number of articles: "; cin >> nrOfArticles;

    // Calculate total price and tax rate

    rateSEK = price \* (RATE / 100); // \*FIXED\* changed from totalPrice to price to calculate rateSEK.

    totalPrice = nrOfArticles \* price \* (1 + RATE/100);

    // Display result with 2 decimals

    cout << fixed << showpoint << setprecision(2);

    cout << nrOfArticles << " number of articles cost " << totalPrice << " kr. " << endl

        << "Of this " << rateSEK \* nrOfArticles << " kr is the tax rate." << endl;

        // \*FIXED\* multiply rateSEK with nrOfArticles to get the total tax rate.

}

## Task 5

I used division and modulo operator to get hours and minutes. I added 0.5 in order to get the integer to round in the right direction.

//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

// File: Source.cpp

// Summary: This program calculates the remaining traveltime while driving

// Version: 1.1

// Owner: Jesper Malmberg

//-----------------------------------------------------------------------

// Log: 2021-11-15 Created file

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

using namespace std;

// Prototypes

void travelTimeCalculation();

int main() {

    char answer;

    do {

        travelTimeCalculation();

        cout << "Calculate one more time? (Y/N): ";

        cin >> answer;

    } while (answer == 'Y' || answer == 'y');

    return 0;

}

//-----------------------------------------------------------------

// void travelTimeCalculation()

//-----------------------------------------------------------------

void travelTimeCalculation()

{

    double miles = 0;           // The remaining travel distance in Swedish miles (1 mile = 10km).

    double averageSpeed = 0;    // The average speed for the remiander of the trip.

    cout << "Enter remaining travel distance in Swedish miles: ";

    cin >> miles;

    cout << "Enter the average speed in km/h: ";

    cin >> averageSpeed;

    double travelTime = miles \* 10 \* 60 / averageSpeed;

    int hours = travelTime / 60;

    int min = travelTime + 0.5; // Add 0.5 in order to get correct rounding of integers.

    int minutes = min % 60;

    //cout << fixed << showpoint << setprecision(2);

    cout << "Remaining travel time: " << hours <<" hours and " << minutes << " minutes." << endl;

}

## Task 6

I assumed 1 level was 3 dB, so in the printout 3 levels of increase of 3 dB is a total increase of 9 dB.

//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

// File: Source.cpp

// Summary: This program calculates the power vs dB increase

// Version: 1.1

// Owner: Jesper Malmberg

//-----------------------------------------------------------------------

// Log: 2021-11-18 Created file

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

using namespace std;

// Prototypes

void dBCalculation();

int main() {

    char answer;

    do {

        dBCalculation();

        cout << "One more time? (Y/N): ";

        cin >> answer;

    } while (answer == 'Y' || answer == 'y');

    return 0;

}

//-----------------------------------------------------------------

// void dBCalculation()

// Calculates the decibel and level increase when increassing power

//-----------------------------------------------------------------

void dBCalculation() {

    double p0, p1, dB;  // p0 is reference W, p1 is incresed W, dB is decibel.

    int level;          // The sound level increase.

    cout << "POWER CALCULATIONS" << "\n";

    cout << "==================" << "\n";

    cout << "Sound Power P0: ";

    cin >> p0;

    cout << "Increased Sound Power P1: ";

    cin >> p1;

    cout << "==================" << endl;

    // Make sure the incresed power is larger than the reference power

    if (p0 > p1) {

        cout << "Please enter a P1 that is larger than P0";

    }

    else {

        dB = 10 \* log10(p1 / p0);

        level = dB / 3;

        cout << "A change in power from " << fixed << setprecision(1) << p0 << " to " << p1 << " corresponds to " << dB << " dB" << endl;

        cout << "That is, " << level << " level of 3 dB";

    }

    cout << endl;

}

## Task 7

Again, I used division and modulo operator to divide the 3 digit number into three parts to later add them up. I also added an input check to make sure only 3 digit numbers were manipulated. This was achieved with a helper function that I named numberLength.

//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

// File: Source.cpp

// Summary: This program adds the individual digits of a 3 digit number

// Version: 1.1

// Owner: Jesper Malmberg

//-----------------------------------------------------------------------

// Log: 2021-11-19 Created file

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

#include <typeinfo>

using namespace std;

// Prototypes

void numberAddition();

int numberLength(int);

int main() {

    char answer;

    do {

        numberAddition();

        cout << "One more time? (Y/N): ";

        cin >> answer;

    } while (answer == 'Y' || answer == 'y');

    return 0;

}

//-----------------------------------------------------------------

// void numberAddition()

//-----------------------------------------------------------------

void numberAddition() {

    int number;

    int first, second, third;

    cout << "Enter a three-digit number: ";

    cin >> number;

    // Check that number entered is exactly 3 digits.

    if (numberLength(number) != 3) {

        cout << "Make sure the number entered is exactly 3 digits...";

        cout << endl;

        numberAddition(); // Run again if incorrect number entered.

    }

    else {

        // Split three digit number into individual digits

        first = number / 100;

        second = (number % 100) / 10;

        third = number % 10;

        cout << "The sum of the three digits is: " << first + second + third;

        cout << endl;

    }

}

//-----------------------------------------------------------------

// int numberLenght()

// Checks the length of an entered int

// Returns an int

//-----------------------------------------------------------------

int numberLength(int number) {

    int counter = 0;

    while (number) {

        number = number / 10;

        counter++;

    }

    return counter;

}

## Task 8

I used an include of <cmath> to use the pow() operator for the windchill formula.

//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

// File: Source.cpp

// Summary: This program calculates the wind chill factor.

// Version: 1.1

// Owner: Jesper Malmberg

//-----------------------------------------------------------------------

// Log: 2021-11-19 Created file

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

#include<cmath> // Used for math pow()

using namespace std;

// Prototypes

void windChillFactor();

int main() {

    char answer;

    do {

        windChillFactor();

        cout << "One more time? (Y/N): ";

        cin >> answer;

    } while (answer == 'Y' || answer == 'y');

    return 0;

}

//-----------------------------------------------------------------

// void windChillFactor()

// This function takes temperature in C and windspeed in m/s and

// calculates the wind chill factor

//-----------------------------------------------------------------

void windChillFactor() {

    double temperature; // Temperature in C

    double windSpeed;   // Windspeed in m/s

    double chillEffect; // The calculated chill effect

    cout << "Enter the temperature in C: ";

    cin >> temperature;

    cout << "Enter the windspeed in m/s: ";

    cin >> windSpeed;

    // This formula calculates the wind chill factor

    chillEffect = 13.126667 + 0.6215 \* temperature - 13.924748 \* pow(windSpeed, 0.16) + 0.4875195 \* temperature \* pow(windSpeed, 0.16);

        cout << fixed << setprecision(0) << temperature << " C and " << windSpeed << " m/s gives the perceived temperature " << chillEffect << " C";

        cout << endl;

}

## Task 9

Pretty straightforward calculations for investment growth over 4 years.

//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

// File: Source.cpp

// Summary: This program calculates investements with interest

// Version: 1.1

// Owner: Jesper Malmberg

//-----------------------------------------------------------------------

// Log: 2021-11-19 Created file

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

using namespace std;

// Prototypes

void investmentGrowth();

int main() {

    char answer;

    do {

        investmentGrowth();

        cout << "One more time? (Y/N): ";

        cin >> answer;

    } while (answer == 'Y' || answer == 'y');

    return 0;

}

//-----------------------------------------------------------------

// void investmentGrowth

// Calculates the growth with interest over 4 years

//-----------------------------------------------------------------

void investmentGrowth() {

    double investment;                 // The initial investment

    double year1, year2, year3, year4; // The interest every year

    double investmentWithInterest;     // The investement value after 4 years

    cout << "Enter the initial investment in SEK: ";

    cin >> investment;

    cout << "Enter percentage year 1: ";

    cin >> year1;

    cout << "Enter percentage year 2: ";

    cin >> year2;

    cout << "Enter percentage year 3: ";

    cin >> year3;

    cout << "Enter percentage year 4: ";

    cin >> year4;

    // Calculate the investment with compounding interest

    investmentWithInterest = investment \* (1 + year1 / 100);

    investmentWithInterest \*= (1 + year2 / 100);

    investmentWithInterest \*= (1 + year3 / 100);

    investmentWithInterest \*= (1 + year4 / 100);

    cout << fixed << setprecision(2) << "Your investment is now worth: " << investmentWithInterest << " SEK";

    cout << endl;

}

## Task 10

Again, <cmath> was used for the pow() method.

//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

// File: Source.cpp

// Summary: This program calculates compounding interest over years

// Version: 1.1

// Owner: Jesper Malmberg

//-----------------------------------------------------------------------

// Log: 2021-11-19 Created file

//-----------------------------------------------------------------------

// Preprocessor directives

#include <iostream>

#include <iomanip>

#include <cmath> // For math pow()

using namespace std;

// Prototypes

void input();

void compoundInterest();

void print(double, double);

// Global Variables

const double INTEREST = 3;

double initialAmount;

double years;

int main() {

    char answer;

    do {

        input();

        compoundInterest();

        cout << "One more time? (Y/N): ";

        cin >> answer;

        cout << endl;

    } while (answer == 'Y' || answer == 'y');

    return 0;

}

//-----------------------------------------------------------------

// void input

// Takes console input for manipulation

//-----------------------------------------------------------------

void input() {

    cout << "COMPOUND INTEREST\n";

    cout << "==============\n";

    cout << "\n";

    cout << "Load initial amount: ";

    cin >> initialAmount;

    cout << "Load number of years: ";

    cin >> years;

    cout << endl;

}

//-----------------------------------------------------------------

// void compoundInterest

// Calculates the compounding interest over 4 years

//-----------------------------------------------------------------

void compoundInterest() {

    double capital; // The investment with compunding interest

    capital = initialAmount\* pow((1 + INTEREST / 100), years);

    print(years, capital);

}

//-----------------------------------------------------------------

// void print

// Prints manipulated input to conosle

//-----------------------------------------------------------------

void print(double years, double capital) {

    cout << "The capital with " << INTEREST << " % interest rate and after " << 10 << " years will be: " << capital << " SEK";

    cout << endl;

}

## 