

1DV533 STEP 3 Assignment report

Jesper Malmberg em222vs@student.lnu.se

Task 1

I simply created three functions, `getSide()` that processes the input, `cubeVolume()` that calculates the volume and `printVolume()` which prints the result to the terminal window.

```
//-----  
//-----  
// File: Task_1.cpp  
// Summary: This program computes the volume of a cube with specified side  
// Version: 1.1  
// Owner: Jesper Malmberg  
//-----  
// Log: 2021-12-06 Created file  
//-----  
// Preprocessor directives  
#include <iostream>  
#include <cctype> // toupper  
using std::cout;  
using std::cin;  
using std::endl;  
// Prototypes  
void getSide(double&); // Parameter passed as reference  
void cubeVolume(double&, double); // Parameter passed as reference and as value  
void printVolume(double);  
  
int main()  
{  
    double side = 0.0, volume = 0.0;  
    char answer = 'Y';  
    cout << "Cubes" << endl;  
    cout << "====" << endl << endl;  
    do  
    {  
        // Read side  
        getSide(side);  
        // Calculate volume  
        cubeVolume(volume, side);  
        // Print volume  
        printVolume(volume);  
  
        cout << endl << "One more time (Y/N)? ";
```

```

        cin >> answer;
    } while (toupper(answer) == 'Y');
    return 0;
}

//-----
// void getSide()
// Takes the input side length as parameter passed by reference
//-----
void getSide(double &side) {
    cout << "Enter the side length of a cube to be computed: ";
    cin >> side;
}

//-----
// void cubeVolume()
// Calculates the volume of a cube
// Takes side and volume as parameters. Side is passed as value and
// volume by reference
//-----
void cubeVolume(double &volume, double side) {
    volume = side * side * side;
}

//-----
// void printVolume()
// Prints the referenced argument to the console
//-----
void printVolume(double volume) {
    cout << "Volume of the cube is: " << volume << " volume units" << endl;
}

```

Task 2

I created a function, fahrenheitToCelcius that passes two parameters, fahrenheit and celsius. Fahrenheit is passed by value as it is not changing, Celsius was passed by reference as it is calculated in the function. The function also returns a Boolean to indicate whether or not the value is above 0C.

```

//-----
//-----
// File: Task_2.cpp
// Summary: This program converts Fahrenheit to Celcius
// Version: 1.1
// Owner: Jesper Malmberg
//-----

```

```

// Log: 2021-12-06 Created file
//-----
// Preprocessor directives
#include <iostream>
#include <iomanip>
#include <cctype> // toupper
using namespace std;
// Prototypes
bool fahrenheitToCelsius(double, double &);

int main()
{
    double fahrenheit, celsius;
    bool aboveZero;
    char answer = 'Y';
    cout << "Temperature conversion" << endl;
    cout << "===== " << endl << endl;
    do
    {
        // Take the input
        cout << "Enter temperature in Fahrenheit : ";
        cin >> fahrenheit;
        // Make the conversion
        aboveZero = fahrenheitToCelsius(fahrenheit, celsius);
        cout << setprecision(3) << celsius << " degrees Celcius." << endl;
        // using std::boolalpha do print true/false instead of 1 or 0
        cout << "Boolean value returned by the conversion function is: " <<
boolalpha << aboveZero << endl;

        cout << endl << "One more time (Y/N)? ";
        cin >> answer;
    } while (toupper(answer) == 'Y');
    return 0;
}
//-----
// void fahrenheitToCelcius()
// Converts Fahrenheit to Celcius and returns True if above 0 C
//-----
bool fahrenheitToCelsius(double fahrenheit, double& celsius) {
    celsius = (fahrenheit - 32) * 5/9;
    if (celsius > 0) {
        return true;
    }
    return false;
}

```

Task 3

I used a simple for-loop to iterate how many times the base was going to multiply itself. If no value for exp was passed 2 was sent as default.

```
//-----  
//-----  
// File: Task_3.cpp  
// Summary: This program increases x with the power of y  
// Version: 1.1  
// Owner: Jesper Malmberg  
//-----  
// Log: 2021-12-09 Created file  
//-----  
// Preprocessor directives  
#include <iostream>  
  
using namespace std;  
// Prototypes  
long power(long base, long exp = 2); // Default value for exp is declared in  
// Prototype  
  
int main()  
{  
    for (int i = 0; i < 10; i++) {  
        cout << power(2, i) << " "; // prints 2 raised to the power of i  
    }  
  
    cout << endl;  
    cout << power(3) << endl; // will print 9  
    cout << power(4) << endl; // will print 16  
    cout << power(5) << endl; // will print 25  
    return 0;  
}  
//-----  
// long power() this function takes a long number  
// and raises it by the power of defined exp which  
// defaults to 2  
//-----  
long power(long base, long exp) {  
    long result = base;  
    for (int i = 1; i < exp; i++) {  
        result = result * base;  
    }  
    return result;  
}
```

Task 4

Not much to add, I used the provided formula in a function to calculate estimated height.

```
//-----  
//-----  
// File: Task_4.cpp  
// Summary: This program estimates the height of a child with input parameters  
// of the father and the mother  
// Version: 1.1  
// Owner: Jesper Malmberg  
//-----  
// Log: 2021-12-10 Created file  
//-----  
// Preprocessor directives  
#include <iostream>  
#include <cctype>  
  
using namespace std;  
// Prototypes  
int heightEstimate(char sex, double hMother, double hFather);  
  
int main()  
{  
    char sex;  
    double hMother;  
    double hFather;  
    char answer = 'Y';  
    do {  
        cout << "Enter the sex of the child (M/F): ";  
        cin >> sex;  
  
        cout << "Enter the height of the MOTHER in cm: ";  
        cin >> hMother;  
        cout << "Enter the height if the FATHER in cm: ";  
        cin >> hFather;  
        cout << "The estimated height of the child is " << heightEstimate(sex,  
hMother, hFather) << " cm.";  
        cout << endl;  
  
        cout << endl << "One more time (Y/N)? ";  
        cin >> answer;  
    } while (toupper(answer) == 'Y');  
    return 0;  
}
```

```

//-----
// This function estimates the height of a child
// It takes the sex of the child, height of the father and
// the mother as input parameters
//-----
int heightEstimate(char sex, double hMother, double hFather) {

    if (toupper(sex) == 'F') {
        return ((hMother * 13 / 12) + hFather) / 2;

    }
    if (toupper(sex) == 'M') {
        return ((hFather * 12 / 13) + hMother) / 2;
    }
}

```

Task 5

I created 3 functions, print, load and discount. In load I passed the parameter sum by reference as this changes with the added discount.

```

//-----
//-----
// File: Task_5.cpp
// Summary: This program calculates payments with different discounts
// Version: 1.1
// Owner: Jesper Malmberg
//-----
// Log: 2021-12-10 Created file
//-----
// Preprocessor directives
#include <iostream>
#include <iomanip>

using namespace std;
// Function prototypes
void print(double sum, double discountKr);
void load(double &sum);
double discount(double sum);

int main()
{
    double sum = 0.0, discountKr = 0.0;
}

```

```

char answer;
do {
    printf("\033c"); // Resets the terminal window

    cout << "Discount" << endl;
    cout << "=====" << endl << endl;
    // Load the input
    load(sum);

    discountKr = sum * discount(sum) / 100;

    // The sum after discount is taken out
    sum = sum - discountKr;

    // Print to the terminal
    print(sum, discountKr);

    cout << endl << "One more time (Y/N)? ";
    cin >> answer;
} while (toupper(answer) == 'Y');
return 0;
}

//-----
// print() This function prints the results to the terminal window
// It rounds the double to the nearest value
//-----
void print(double sum, double dKr) {
    cout << fixed << setprecision(2);
    cout << "-----" << endl;
    cout.width(24); cout << left << "Discount      : " << right << round(dKr) <<
endl;
    cout.width(24); cout << left << "Net sum      : " << right << round(sum) <<
endl;
    cout << "-----";
}

//-----
// load() This function loads the input from the user
//-----
void load(double &sum) {
    cout << "Enter the sum: ";
    cin >> sum;
}

```

```
//-----
// discount() This function returns the discount depending on the total sum
//-----
double discount(double sum) {
    if (sum <= 500) {
        return 0;
    }
    if (sum > 500 && sum <= 1000) {
        return 5;
    }
    if (sum > 1000 && sum <= 5000) {
        return 10;
    }
    else {
        return 15;
    }
}
```

Task 6

5 functions total, one for each arithmetic operator and an additional to calculate the greatest common demominator. I used a switch statement to input selections from the menu. I used a dummy variable called ignoreInput to get rid of the “/” character from the user’s input.

```
//-----
//-----
// File: Task_6.cpp
// Summary: This program includes 4 basic arithmetic operations on
//          fractional numbers
// Version: 1.1
// Owner: Jesper Malmberg
//-----
// Log: 2021-12-11 Created file
//-----
// Preprocessor directives
#include <iostream>
#include <iomanip>

using namespace std;
// Function prototypes
void addition(int, int, int, int);
void subtraction(int, int, int, int);
void multiplication(int, int, int, int);
```



```

void division(int, int, int, int);
//void printFraction(int, int);
int gcdCalculator(int, int);

int main()
{
    char answer;                                // Continue or stop
program variable
    int numerator1, denominator1, numerator2, denominator2; // The four numbers
of the fraction
    char ignoreInput;                            // Dummy variable to
discard the '/' character
    int option;                                  // Variable for the
selection of arithmetic operation

    do {
        printf("\033c"); // Resets the terminal window

        cout << "Enter the first fractional number (x/y): ";
        cin >> numerator1 >> ignoreInput >> denominator1;
        cout << "Enter the second fractional number (x/y): ";
        cin >> numerator2 >> ignoreInput >> denominator2;
        cout << endl;

        // Menu option
        cout << "Which arithmetic operation would you like to perform? Enter the
number: " << endl;
        cout << "(1) Addition          " << numerator1 << "/" << denominator1 << " +
" << numerator2 << "/" << denominator2 << endl;
        cout << "(2) Subtraction       " << numerator1 << "/" << denominator1 << " -
" << numerator2 << "/" << denominator2 << endl;
        cout << "(3) Multiplication  " << numerator1 << "/" << denominator1 << " *
" << numerator2 << "/" << denominator2 << endl;
        cout << "(4) Division          " << numerator1 << "/" << denominator1 << " /
" << numerator2 << "/" << denominator2 << endl;
        cin >> option;

        // Switch for the menu
        switch (option) {
            case 1:
                addition(numerator1,denominator1, numerator2, denominator2);
                break;
            case 2:
                subtraction(numerator1, denominator1, numerator2, denominator2);
                break;

```

```

        case 3:
            multiplication(numerator1, denominator1, numerator2,
denominator2);
            break;
        case 4:
            division(numerator1, denominator1, numerator2, denominator2);
            break;
    }
    cout << endl;
    cout << "One more time (Y/N)?";
    cin >> answer;

    } while (toupper(answer) == 'Y');
    return 0;
}

//-----
// addition()
// This function takes 4 integers representing to fractional numbers
// and calculates the sum of the two
//-----
void addition(int num1, int den1, int num2, int den2) {
    int newNumerator = num1 * den2 + num2 * den1;
    int newDenominator = den1 * den2;

    int gCd = gcdCalculator(newNumerator, newDenominator);
    cout << "The sum of " << num1 << "/" << den1 << " + " << num2 << "/" << den2
<< " = " << newNumerator / gCd << "/" << newDenominator / gCd;

}

//-----
// subtraction()
// This method takes 4 integeres representing 2 fractional numbers
// and calculates the difference between the two
//-----
void subtraction(int num1, int den1, int num2, int den2) {
    int newNumerator = num1 * den2 - num2 * den1;
    int newDenominator = den1 * den2;

    int gCd = gcdCalculator(newNumerator, newDenominator);
    cout << "The difference between " << num1 << "/" << den1 << " - " << num2 <<
"/" << den2 << " = " << newNumerator / gCd << "/" << newDenominator / gCd;
}

//-----

```

```

// multiplication()
// This function takes 4 integers representing 2 fractional numbers
// and calculates the product of the two
//-----
void multiplication(int num1, int den1, int num2, int den2) {
    int newNumerator = num1 * num2;
    int newDenominator = den1 * den2;

    int gCd = gcdCalculator(newNumerator, newDenominator);
    cout << "The product of " << num1 << "/" << den1 << " * " << num2 << "/" <<
den2 << " = " << newNumerator / gCd << "/" << newDenominator / gCd;
}

//-----
// division()
// This function takes 4 integers representing 2 fractional numbers
// and calculates the quotient of the two
//-----
void division(int num1, int den1, int num2, int den2) {
    int newNumerator = num1 * den2;
    int newDenominator = den1 * num2;

    int gCd = gcdCalculator(newNumerator, newDenominator);
    cout << "The quotient of " << num1 << "/" << den1 << " / " << num2 << "/" <<
den2 << " = " << newNumerator / gCd << "/" << newDenominator / gCd;
}

//-----
// gcdCalculator()
// This helper function takes two integers and calculates the greatest common
// denominator. It returns the gcd or 1 if there is none.
//-----
int gcdCalculator(int num1, int num2) {

    // If the fractional number is negative change it to positive in order to
    find gcd
    if (num1 < 0) {
        num1 = -num1;
    }
    if (num2 < 0) {
        num2 = -num2;
    }

    // Default value, if no common denominator, return 1
    int gcd = 1;

```

```

    for (int i = 1; i <= num1 && i <= num2; i++)
    {
        if (num1 % i == 0 && num2 % i == 0)
            gcd = i;
    }
    return gcd;
}

```

Task 7

Booth leapYear and daysInMonth functions are pretty simple with if statements to define the logic path of the program. The tomorrow function however was a bit trickier to figure out how to use / and % to break out the different parts of the input date. It works well with the other two functions to calculate the next day date. There is not verification of correct input from the user however.

```

//-----
//-----
// File: Task_7.cpp
// Summary: This program checks the various properties of entered dates
// Version: 1.1
// Owner: Jesper Malmberg
//-----
// Log: 2021-12-11 Created file
//-----
// Preprocessor directives
#include <iostream>
#include <iomanip>

using namespace std;
// Function prototypes
bool leapYear(int);
int daysInMonth(int, int);
int tomorrow(int);

int main()
{
    char answer;    // Variable to keep program running
    int selection;  // Selection from the menu
    int year;       // Year as input by user
    int month;      // Month as input by user
    int date;       // Day/Date as input by user

    do {

```

```

printf("\033c"); // Resets the terminal window

cout << "What would you like to do?" << endl;
cout << "(1) Check if leap year" << endl;
cout << "(2) Check number of days in a month" << endl;
cout << "(3) Check tomorrow's date" << endl;
cin >> selection;

switch (selection) {
case 1:
    cout << "Enter a year: ";
    cin >> year;
    if (leapYear(year))
        cout << "This is a leap year!" << endl;
    else
        cout << "This is not a leap year!" << endl;
    break;
case 2:
    cout << "Enter the year and month with a space between (2004 5)" <<
endl;

    cin >> year >> month;
    cout << "The number of days in this month is: " << daysInMonth(year,
month);

    break;
case 3:
    cout << "Enter a date on the form YYYYMMDD: ";
    cin >> date;
    cout << "Tomorrow it is: " << tomorrow(date);
    break;
}

    cout << endl << "One more time (Y/N)? ";
    cin >> answer;
} while (toupper(answer) == 'Y');
return 0;
}

//-----
// leapYear()
// This function takes a year as an input parameter and returns true or
// false depending if said year is a leapyear
//-----
bool leapYear(int year) {
    if(year % 4 == 0 && year % 100 != 0) {
        return true;
    }
}

```

```

    }
    else if(year % 400 == 0) {
        return true;
    }
    // Default return
    return false;
}

//-----
// daysInMonth()
// This function takes the year abd month as integers and returns the
// number of days in the specified month
//-----
int daysInMonth(int year, int month) {
    // April June September and November has 30 days
    if (month == 4 || month == 6 || month == 9 || month == 11) {
        return 30;
    }
    // Check if February
    if (month == 2) {
        // If also leapyear
        if (leapYear(year)) {
            return 29;
        }
        else {
            return 28;
        }
    }
    // All other months
    return 31;
}

//-----
// tomorrow()
// This function takes a date in the format YYYYMMDD and returns tomorrow's
// date
//-----
int tomorrow(int date) {
    // Split date into Year, Month, Day
    int day = date % 100;
    int month = date / 100 % 100;
    int year = date / 10000;

    // If it's the last day of the month
    if (day == daysInMonth(year, month)) {

```

```
// Check if month is December
if (month == 12) {
    return (year + 1) * 10000 + 1 * 100 + 1;
}
// Any other month than December
return year * 10000 + (month + 1) * 100 + 1;
}
// Default return, if it's not the last day of the month
return year * 10000 + month * 100 + day + 1;
}
```

Task 8

I did not have time to complete this one unfortunately.