**CSE1502 – Intro to Software Dev. with C++**

**Software Development Lab Report**

**Fall 2021**

*By*

Timothy Calco

*Mechanical Engineering*

*Tcalco2021@my.fit.edu*

Oct 3, 2021

Homework Problem Set #1

Teaching Instructor:

Stefan Joe-Yen, Ph.D.

# 1. Problem Statement

*Soda Dispenser: The Program simulates a soda machine by operating in a loop that accepts coin inputs until total input reaches or exceeds $1, at which point the program calculates the remainder and “outputs” the remainder in coins. The machine must reject any input other than 5, 10, and 25.*

*ASCII Shapes: Program must draw ASCII Shapes in console and operate in a loop. User should be able to choose between 4 different shapes, either rectangles or triangles, filled or empty. The program must request a height and width for the rectangles, and must request a height for the triangles.*

*ASCII Checkerboard Flag: The Program must dynamically (that is, through code command structures and loops, and NOT by simply outputting the final product) output a checkerboard flag in the console. The only outputs that are allowed are cout << endl; cout << “\* “; and cout << “ “;. These must be placed in loops and statements that draw out a checkerboard pattern.*

# 2. Requirements

*This section lists the functional requirements given for the software. Functional requirements dictate what the software needs to do (i.e., that major functions of the software). Each requirement must be presented using a numbered list. In addition, each requirement must specify one (and only one) function of the system, it must be attainable and verifiable.*

*Soda Dispenser:*

1. *The software shall display a message upon being run*
2. *The software shall prompt for user input*
3. *The software shall add user input to a value, or reject invalid input*
4. *The software shall stop accepting user input when a value is reached*
5. *The software shall determine “change” by dividing the remainder by preset values.*
6. *The software shall terminate.*

*ASCII Shapes:*

1. *The Software shall operate in a loop.*
2. *The Program shall prompt for, and take, user input*
3. *Based on user input, the software shall execute conditional code blocks.*
4. *Each individual code block shall represent a shape*
   1. *Filled Rectangle: Prompt user for height and width, output appropriate shape*
   2. *Empty Rectangle: Prompt user for height and width, output appropriate shape*
   3. *Filled Triangle: Prompt user for height, output appropriate shape*
   4. *Empty Triangle: Prompt user for height, output appropriate shape*

*ASCII Checkerboard Flag:*

1. *The Software shall operate in a command structure*
2. *The software can only output “ “, “\* “, or end the line.*
3. *It must use the command structures to dynamically output alternating input patterns.*

# 3. Software Construction (C++ Code)

*Soda Dispenser:*

#include <iostream>

#include <cmath>

#include <string>

using namespace std;

int main() {

//Declaring Variables

int i;

int nickel = 5;

int dime = 10;

int quarter = 25;

int x = 0;

do {

//Get input

cout << "Soda costs $1. Please input a coin." << endl;

cin >> i;

if (i == quarter) {

x += 25;

cout << "Total input: " << x << " cents" << endl;

//if 25 is input, 25 is added to total input

}

else if (i == dime) {

x += 10;

cout << "Total input: " << x << " cents" << endl;

//if 10 is input, 10 is added to total input

}

else if (i == nickel) {

x += 5;

cout << "Total input: " << x << " cents" << endl;

//if 5 is input, 5 is added to total input

}

else

cout << "Invalid input. Try again." << endl;

} while (x < 100);

//Prepare Variables for Change Calculation

x = x - 100;

int quarter\_change;

int dime\_change;

int nickel\_change;

//Find, Then Remove Quarters Change

quarter\_change = x / 25;

x = x - (quarter\_change \* 25);

//Find, Then Remove Dimes Change

dime\_change = x / 10;

x = x - (dime\_change \* 10);

//Find, Then Remove Nickels Change

nickel\_change = x / 5;

x = x - (nickel\_change \* 5);

//Final Output

cout << "\nEnjoy your soda." << endl;

cout << "Don't forget your change.\n" << endl;

//Change Output

printf("Quarters: %d \nDimes: %d \nNickels: %d",

quarter\_change, dime\_change, nickel\_change);

cout << "\n\n";

}

*ASCII Shapes:*

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

int main() {

//Declare Variables

int length;

int height;

int x;

do {

//First Output (List Options)

cout << "\nSelect a shape from the following options.\n" << endl;

cout << "(1) Filled Rectangle" << endl;

cout << "(2) Empty Rectangle" << endl;

cout << "(3) Filled Triangle" << endl;

cout << "(4) Empty Triangle" << endl;

cout << "(5) Exit\n" << endl;

cin >> x;

if (x == 1) {

//Filled Rectangle

cout << "Enter the height of your rectangle." << endl;

cin >> height;

cout << "Please enter the length of your rectangle" << endl;

cin >> length;

cout << endl;

for (int i = 0; i < height; i++)

{

for (int i = 0; i < length; i++)

{

cout << "\* ";

//Each row outputs (length) "\* "'s

}

cout << endl;

//Outputs (height) rows

}

}

else if (x == 2) {

//Empty Rectangle

cout << "Please enter the height of your rectangle.\n";

cin >> height;

cout << "Please enter the length of your rectangle\n";

cin >> length;

cout << endl;

//First Line

for (int i = 0; i < length; i++)

{

cout << "\* ";

//Outputs "\* " (length) times

}

cout << endl;

//Middle Lines

for (int i = 1; i < (height - 1); i++)

{

cout << "\* "; //First output "\* "

for (int i = 1; i < (length - 1); i++)

{

cout << " ";

//After first and until the last output are " "

}

cout << "\* " << endl; //End line with "\* "

}

//Last Line

for (int i = 0; i < length; i++)

{

cout << "\* ";

//Outputs "\* " (length) times

}

cout << endl;

}

else if (x == 3) {

//Filled Triangle

cout << "Please enter the height of your triangle." << endl;

cin >> height;

for (int a = 0; a <= height; a++)

{

for (int b = 0; b < a; b++)

{

cout << "\* ";

//Outputs "\* " a times, a at this point is the height of that line

}

cout << endl;

//I could Initialize a as 1 but I like that it goes straight to the endl; first

}

}

else if (x == 4) {

//Empty Triangle

cout << "Please enter the height of your triangle" << endl;

cin >> height;

cout << endl;

//First Two Lines

for (int a = 1; a <= 2 && a <= height; a++)

{

for (int b = 0; b < a; b++)

{

cout << "\* ";

//Outputs \* once for line one, twice for line 2

}

cout << endl;

//This loops until a = 2 or height.

}

//Lines 3 Thru (Height - 1)

for (int a = 3; a < height; a++)

{

for (int b = 0; b < a; b++) //Outputs (a) times

{

if (b == 0 || b == (a - 1))

cout << "\* ";

//First and last outputs of a line are "\* "

else

cout << " ";

//Middle outputs are " "

}

cout << endl;

//a starts at 3, ends at line before height is reached

}

//Last Line

for (int b = 0; b < height; b++)

{

cout << "\* ";

//Outputs "\* " (height) times

}

cout << endl;

}

else if (x == 5) {

cout << "Terminating Program...\n" << endl;

break;

//Ends program by break;'ing out of the loop

}

else {

cout << "Invalid Input. Please select options 1 thru 5." << endl;

//Any invalid input should direct here.

}

} while (true); //Keeps the program looping until loop is broken

}

*ASCII Checkerboard Flag:*

#include <iostream>

using namespace std;

int main () {

//Declare Variables

int height = 7; //7 rows in instructions

int width = 16; //16 outputs per row

int indent = 1; //indent%2=1 prints "\* ", indent%2=0 prints " "

for (int r = 0; r < height; r++) //Rows height

{

for (int c = 0; c < width; c++) //Outputs per line

{

if (indent % 2 == 0) {

cout << " ";

indent++;

//this output is the " "

}

else {

cout << "\* ";

indent++;

//this output is the "\* "

}

}

cout << endl; //single endl here makes it more compact vertically.

indent++; //Each new row changes indent

}

}

# 4. Software Testing

*Soda Dispenser:*

*Text

Description automatically generated*

*ASCII Shapes:*

*Text

Description automatically generated*

*Continued on next page.*

*Text

Description automatically generated*

*ASCII Flag:*

*Text

Description automatically generated*

# 5. Self-Reflection

*Soda Dispenser:*

*This program was logically very simple. I structured it slightly different from how I usually do, since I declared some of the variables later on in the code, rather than all at the top. They weren’t used by the initial loop so I declared them after. I had fun programming this, and didn’t encounter any challenges.*

*ASCII Shapes:  
This was really challenging. I struggled with each shape, and had a hard time wrapping my head around the basic algorithms and logic of each shape. Eventually I had a bit of a breakthrough in realizing I could break the shapes up into composite parts, as in the case of the empty shapes. That made it way easier and it was smooth sailing from then, but it took me a long time to get there.*

*ASCII Checkerboard Flag:*

*After the ASCII Shapes, this one was a breeze. I chose to base the indents on an external variable, adding to it twice in a loop, alternating with each horizontal output and with each row. I am aware that the code only creates a checkerboard if the width is even, that is the nature of my code structure. If it were odd, I would need to remove the last “indent++”.*

*Conslusion:*

*I enjoyed this assignment, but I started late because of my personal situation. I learned a lot, mostly about for loops, and I think that I’m better prepared to tackle other coding problems.*