#include <iostream>

#include <iomanip>

#include <string>

#include <stdlib.h>

using namespace std;

int GetValue(string type, int min, int max);

void PrintRectangle(int width, int height);

int main()

{

// Declare Variables

int width, height, min = 0, max = 25;

string type;

// Request user input

cout << endl << endl;

cout << "Drawing a Rectangle" << endl;

cout << endl << endl;

type = "width";

width = GetValue(type, min, max); // Get width using user defined function GetValue

type = "height";

height = GetValue(type, min, max); // Get height using user defined function GetValue

system("CLS"); // Clear Screen

cout << endl << endl;

// Print results

PrintRectangle(width, height); // Print Rectangle using void function PrintRectangle

cout << endl;

exit (0);

}

// User Defined Value-Returning Function GetValue:

// Get user input (string) and check that it contains only numerals, convert string value to int, and test that

// int value is greater than or equal to zero. Print error statement and get new user input if invalid.

int GetValue(string type, int min, int max)

{

// Declare Variables

string str\_value;

char first;

int retest, sign, StringLength, count, position, count2, position2, value, error;

retest = 0; // Reset Counters

StringLength = 0;

while (retest == 0) // Loop A (Get End Value and Test)

{

cout << "Enter the "<< type << " of the rectangle you wish to draw: ";// Ask user for appropriate information

cin >> str\_value; // Get user entered value and determine string length

cout << endl;

first = str\_value.at(0);

sign = 1; // Set/reset positive "value"

if ((first < 46) && (first > 44)) // Loop A1 (Remove leading '-' sign)

{

str\_value.erase(0, 1);

sign = -1; // Set variable to make final int negative

first = 0;

}

StringLength = (str\_value.length());

count = 0; // Set/reset count and position for validity test

position = 0;

while ((count < (StringLength))) // Loop A2 (Tests each character for numeric content)

{

position = str\_value.at(count);

if (isdigit(position)) // Loop A2a (If Else)

++count; // Adds one to count variable; evaluate next character of string

else

break; // Breaks loop A2 when non numeric value is returned (including " ", ".", and "-")

// Count at exit = number of digits found before invalid character

}

if (count == (StringLength)) // Loop A3 (If Else)

{ // Only enter loop if all characters in string are numeric

count2 = 0; // Set/reset counters & variable for conversion

position2 = 0;

value = 0;

error = 0; // Set/reset error flag

while ((count2 < (StringLength))) // Loop A3a (Convert characters in string to integer)

{

position2 = str\_value.at(count2);

value = (value \* 10) + (position2 - 48); // Multiplies previous end by 10 and adds numeric value of ASCII char

++count2; // Adds one to count, allows next character of string to be read

}

}

else

{

error = 3;

value = 0;

}

value = (value \* sign); // Returns trimmed '-' sign from string to int value

if ((value > min) && (value < max)) // Loop A4 (Checks for valid numeric value)

{

retest = 1;

}

if (retest == 0) // Loop A5 (Prints error statement ande returns to loop A)

{

if ((value <= min) && (!(error == 3))) // Loop A5a (Error # to small)

cout << "You have entered an invalid " << type << ". Please enter an integer greater than " << min << ".";

if ((value >= max) && (!(error == 3))) // Loop A5b (Error # to large)

cout << "You have entered an invalid " << type << ". Please enter an integer less than " << max << ".";

if (error == 3) // Loop A5c (Other error)

cout << "You have entered an invalid " << type << ". Please try again";

cout << endl << endl;

cin.clear(); // Reset cin for new user input in loop A

cin.ignore(1000, '\n');

}

if (retest == 1) // Loop A6 return value to main

{

return value;

}

}

}

// User Defined Void Function PrintRectangle

// Takes width and height from GetValue Function and prints a rectangle

void PrintRectangle(int width, int height)

{

// Declare variables

int count = 0;

// Print

cout << "Here is your " << width << " x " << height << " rectangle:" << endl << endl;

for (height = height; height > 0; height--)

{

for (count = width; count > 0; count--)

cout << "X ";

cout << endl;

}

}