// =======================

// Attached: HW #4

// =======================

// HW #4

// =======================

// Youssef Abdelwahab

// CS 1B

// =======================

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

struct Box

{

int id;

float Width;

float Height;

float Length;

Box\* next;

};

typedef Box\* ptrType;

bool insertbox(ptrType& head);

void deleteboxes(ptrType& head);

void displayboxes(const ptrType head);

int main()

{

char answer;

do {

system("cls");

ptrType head = NULL;

cout << "Enter a box (-1 to quit): \n\n";

bool doAgain = true;

do

{

doAgain = insertbox(head);

} while (doAgain == true);

cout << " " << endl;

cout << "Display all records Y/N? ";

cin >> answer;

if (toupper(answer) == 'Y')

system("cls");

displayboxes(head);

if (head != NULL)

{

cout << "delete box? (Y/N)...";

cin >> answer;

if (toupper(answer) == 'Y')

{

system("cls");

deleteboxes(head);

}

}

system("cls");

cout << "Here is the list of boxes after deleting: " << endl;

cout << " " << endl;

displayboxes(head);

cout << " " << endl;

cout << "Do again Y/N? ";

cin >> answer;

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

return 0;

}

// ==== insertbox function ==================================

bool insertbox(ptrType& head)

{

ptrType temp = new(Box);

bool doAgain = true;

int idNum = 0;

cout << setw(10) << "ID: ";

cin >> idNum;

if (idNum == -1)

doAgain = false;

else

{

temp->id = idNum;

cin.ignore();

cout << setw(10) << "Width: ";

cin >> temp->Width;

cout << setw(10) << "Height: ";

cin >> temp->Height;

cout << setw(10) << "Length: ";

cin >> temp->Length;

cout << endl;

doAgain = true;

temp->next = head;

head = temp;

}

return doAgain;

}

// =============================================================

// ==== displayboxed function ================================

void displayboxes(const ptrType head)

{

if (head == NULL)

{

cout << "List is empty.\n";

exit(0);

}

ptrType temp = head;

while (temp != NULL)

{

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

cout << setw(10) << "ID#:" << setw(20) << temp->id << endl;

cout << " " << endl;

cout << setw(10) << "Width:" << setw(20) << temp->Width << endl;

cout << " " << endl;

cout << setw(10) << "Height:" << setw(20) << temp->Height << endl;

cout << " " << endl;

cout << setw(10) << "Length:" << setw(20) << temp->Length<< endl;

cout << " " << endl;

temp = temp->next;

}

}

// =============================================================

// =========deleteboxes=========================================

void deleteboxes(ptrType& head)

{

Box\* lead = head;

Box\* follow = head;

int idNum;

cout << "Enter the ID of the song to be deleted: ";

cin >> idNum;

while (lead->id != idNum && lead->next != NULL)

{

follow = lead;

lead = lead->next;

}

if (lead == head)

{

if (lead->id == idNum)

{

head = head->next;

delete lead;

}

else

{

cout << idNum << " is not found in the list.\n\n";

}

}

else if (lead->next == NULL)

{

if (lead->id == idNum)

{

follow->next = lead->next;

delete lead;

}

else

{

cout << "ID#" << idNum << " is not found in the list.\n\n";

}

}

else

{

follow->next = lead->next;

delete lead;

}

}

Text

Description automatically generatedText

Description automatically generated with low confidence

Graphical user interface, text, application, chat or text message

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated with low confidence

Text

Description automatically generated

Text

Description automatically generated