Question 4: (Final 2015/2016)

Code:

```
#include<iostream>
using namespace std;

const int max_len = 100;

class Car {
public:
	int ID;
	int Wheels;
	double Weight;
	char Type;
};

class Truck {
public:
	int ID;
	int length;
	int weight;
	double Payload;
};

class Vehicle {
public:
	union a {
		Car c;
		Truck t;
		a(){
			c.ID = 0; c.Wheels = 0; c.Weight = 0; c.Type = 'A';
			t.ID = 0; t.length = 0; t.weight = 0; t.Payload = 0;
		}
	}vehicle;
};

class stack {
public:
	int top;
	char s[max_len];
};

void reset(stack *stk) {
	stk->top = 0;
}
void push(stack *stk, char c) {
	stk->top++;
	stk->s[stk->top] = c;
}
char pop(stack *stk) {
	return(stk->s[stk->top--]);
}
bool full(stack *stk) {
	return(stk->top == max_len);
}
bool empty(stack *stk) {
	return(stk->top == 0);
}
```

```
class Node {
public:
	Vehicle info;
	Node *next;
};

void main() {
	char c;
	Vehicle v;
	int countV = 0, countC = 0, countT = 0;
	Node *list, *temp;
	list = new Node;
	cout << "If you want to enter car data press 0\nIf you want to enter truck data
enter 1" << endl;
	cin >> c;
	switch (c) {
	case '0':
		cout << "Enter ID: "; cin >> v.vehicle.c.ID;
		cout << "Enter Wheels: "; cin >> v.vehicle.c.Wheels;
		cout << "Enter Weight: "; cin >> v.vehicle.c.Weight;
		cout << "Enter Type: "; cin >> v.vehicle.c.Type;
		countC++;
		countV++;
		break;
	case '1':
		cout << "Enter ID: "; cin >> v.vehicle.t.ID;
		cout << "Enter length: "; cin >> v.vehicle.t.length;
		cout << "Enter weight: "; cin >> v.vehicle.t.weight;
		cout << "Enter Payload: "; cin >> v.vehicle.t.Payload;
		countT++;
		countV++;
		break;
	}
	list->info = v;
	list->next = NULL;
	temp = list;
	cout << "If you want to continue press y" << endl;
	cin >> c;
	while (c == 'y') {
	again:
		cout << "If you want to enter car data press 0\nIf you want to enter truck
data enter 1" << endl;
		cin >> c;
		switch (c) {
		case '0':
			cout << "Enter ID: "; cin >> v.vehicle.c.ID;
			cout << "Enter Wheels: "; cin >> v.vehicle.c.Wheels;
			cout << "Enter Weight: "; cin >> v.vehicle.c.Weight;
			cout << "Enter Type: "; cin >> v.vehicle.c.Type;
			countC++;
			countV++;
			break;
		case '1':
			cout << "Enter ID: "; cin >> v.vehicle.t.ID;
			cout << "Enter length: "; cin >> v.vehicle.t.length;
			cout << "Enter weight: "; cin >> v.vehicle.t.weight;
```

```
                        cout << "Enter Payload: "; cin >> v.vehicle.t.Payload;
                        countT++;
                        countV++;
                        break;
                default:
                        goto again;
                }
                temp->next = new Node;
                temp = temp->next;
                temp->info = v;
                temp->next = NULL;
                cout << "If you want to continue press y" << endl;
                cin >> c;
        }
        temp = list;
        cout << "Number of cars = " << countC << "\nNumber of trucks = " << countT <<
endl;
        double TotalWeight = 0;
        while (temp != NULL) {
                TotalWeight += temp->info.vehicle.c.Weight + temp->info.vehicle.t.weight;
                temp = temp->next;
        }
        double AverageWeight;
        AverageWeight = TotalWeight / countV;
        cout << "The average weight is " << AverageWeight << endl;
}
```