Source Code

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

struct Courier {
    string name;
    string location;
    string ID;
    float distance;
    float weight;
    int cost;
};

Courier couriers[500];
int couriers_count = 0;

Courier returns[500];
int returns_count = 0;

int failed = 0;

float calculateCost(Courier courier) {
    return (10 * courier.distance/2 * courier.weight/3000) +
70;
}
```

```
string courierID(Courier courier) {
    std::string str_name (courier.name);
    std::string initials (str_name.substr(0,2));
    transform(initials.begin(), initials.end(), initials.begi
n(), ::toupper);
    std::string courier_id = initials + std::to_string(courie
r.distance * courier.weight);
    return courier_id.substr(0,6);
}

Courier bookCourier() {
    string name, location, ID;
    float distance, weight;
    int cost;
    Courier courier;

    cout << "Enter product name: ";
    cin >> courier.name;
    cout << "Enter shipping location: ";
    cin >> courier.location;
    cout << "Enter approx. distance (in kms) from shipping ce
ntre: ";
    cin >> courier.distance;
    cout << "Enter item weight (in grams): ";
    cin >> courier.weight;
    cout << endl;
```

```
    courier.cost = calculateCost(courier);
    courier.ID = courierID(courier);

    couriers[couriers_count] = courier;
    couriers_count += 1;

    cout << "Your courier ID is " << courier.ID << endl;
    return courier;
}

void printDetails(Courier data) {
    cout << endl;
    cout << "Name:     " << data.name << endl;
    cout << "Location: " << data.location << endl;
    cout << "Distance: " << data.distance << " kms" << endl;
    cout << "Weight:   " << data.weight << " grams" << endl;
    cout << "Cost:     " << "Rs. " << data.cost << endl;
    cout << "ID:       " << data.ID << endl;
}

Courier incomingCouriers() {
    int i;
    for (i=0; i<couriers_count; i++) {
        cout << i+1 << ". " << couriers[i].ID << endl;
    }
    cout << endl;
    cout << "Choose a courier ID: ";
    cin >> i;
```

```
    printDetails(couriers[i-1]);
    return couriers[i-1];
}

void removeBooking(int n) {
    for (int i=n; i<couriers_count; ++i) {
        couriers[i] = couriers[i+1];
    }
    couriers_count -= 1;
    failed += 1;
}

void returnCourier() {
    string ID;
    cout << "Enter courier ID: " << endl;
    cin >> ID;
    int found = 0;
    for (int i=0; i<couriers_count; i++) {
        if (couriers[i].ID == ID) {
            cout << "OK. Returning back this courier" << endl
;
            returns[returns_count] = couriers[i];
            removeBooking(i);
            returns_count += 1;
            found = 1;
        }
    }
```

```
    if (!found) {
        cout << "This courier ID was not found!" << endl;
    }
}

Courier incomingReturns() {
    int i;
    for (i=0; i<returns_count; i++) {
        cout << i+1 << ". " << returns[i].ID << endl;
    }
    cout << endl;
    cout << "Choose a return ID: ";
    cin >> i;

    printDetails(couriers[i-1]);
    return returns[i-1];
}

void companyDetails() {
    cout << "Total couriers: " << couriers_count + failed <<
endl;
    cout << "Total returns:  " << returns_count << endl;
}

void introduction() {
    cout << endl << "COURIER MANAGEMENT SYSTEM" << endl << en
dl;
    cout << "Choose an option:" << endl;
```

```
    cout << "1. Book a courier" << endl;
    cout << "2. Get all incoming couriers details" << endl;
    cout << "3. Return a courier" << endl;
    cout << "4. Get all return couriers details" << endl;
    cout << "5. Company details" << endl;
    cout << "6. Exit" << endl;
    cout << endl;
}

int main() {
    int choice;

    while (1) {
        introduction();
        cin >> choice;
        cout << endl;

        switch (choice) {
            case 1:
                bookCourier();
                break;
            case 2:
                incomingCouriers();
                break;
            case 3:
                returnCourier();
                break;
            case 4:
```

```
                incomingReturns();
                break;
            case 5:
                companyDetails();
                break;
            case 6:
                cout << "Exitting" << endl;
                return 0;
            default:
                cout << "Invalid option!" << endl;
                break;
        }
    }

    return 0;
}
```