National University
of computer and emerging sciences

Lab 4

Items	Description
Course Title	Object Oriented Programming
Lab Title	Intro to Structures
Duration	3 Hours
Tools	Eclipse/ C++
Objective	To get familiar with the use of Structures in c++

Introduction to Structures

In C++, a structure (`struct`) is a user-defined data type that groups related variables of different types into a single unit. Structures are useful for organizing complex data and making code more readable.

```
struct StructureName {
   DataType member1;
   DataType member2;
   // More members
};
```

```
struct Person {
    string name;
    int age;
    float height;
};
```

Declaring and Initializing Structures

1 Declaring Structures:

You define a structure with the `struct` keyword followed by the structure's name and its members.

```
struct Book {

    string title;

    string author;

    int pages;

};
```

2 Initializing Structures:

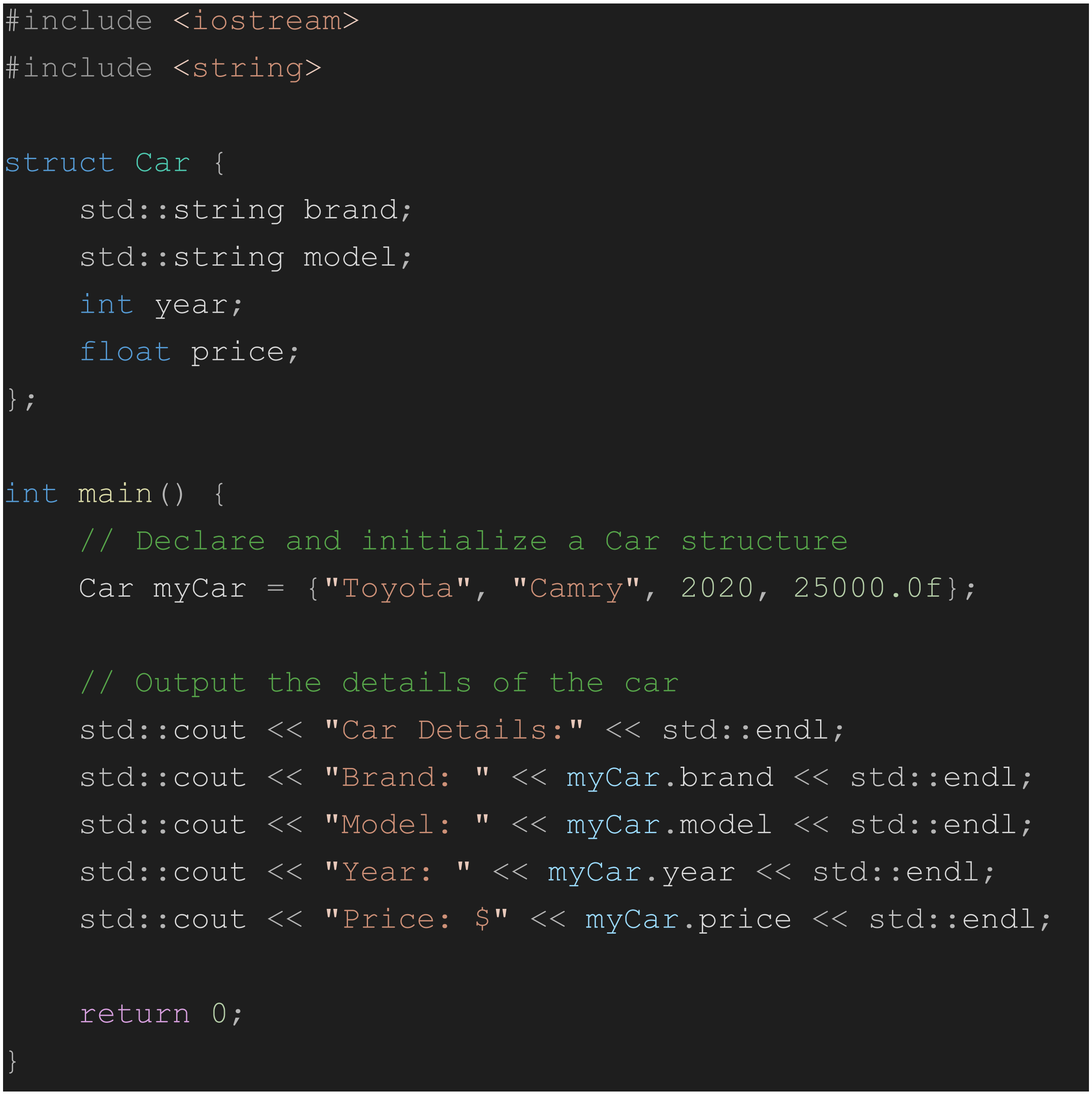
You can initialize a structure when you declare it.

```
Book myBook = {"Object Oriented Programming", "George Orwell", 328};
```

3 Accessing Members:

Use the dot operator (`.`) to access members of a structure.

```
cout << "Title: " << myBook.title << endl;
```



```cpp
#include <iostream>
#include <string>

struct Car {
    std::string brand;
    std::string model;
    int year;
    float price;
};

int main() {
    // Declare and initialize a Car structure
    Car myCar = {"Toyota", "Camry", 2020, 25000.0f};

    // Output the details of the car
    std::cout << "Car Details:" << std::endl;
    std::cout << "Brand: " << myCar.brand << std::endl;
    std::cout << "Model: " << myCar.model << std::endl;
    std::cout << "Year: " << myCar.year << std::endl;
    std::cout << "Price: $" << myCar.price << std::endl;

    return 0;
}
```

Array as a data member in Structure

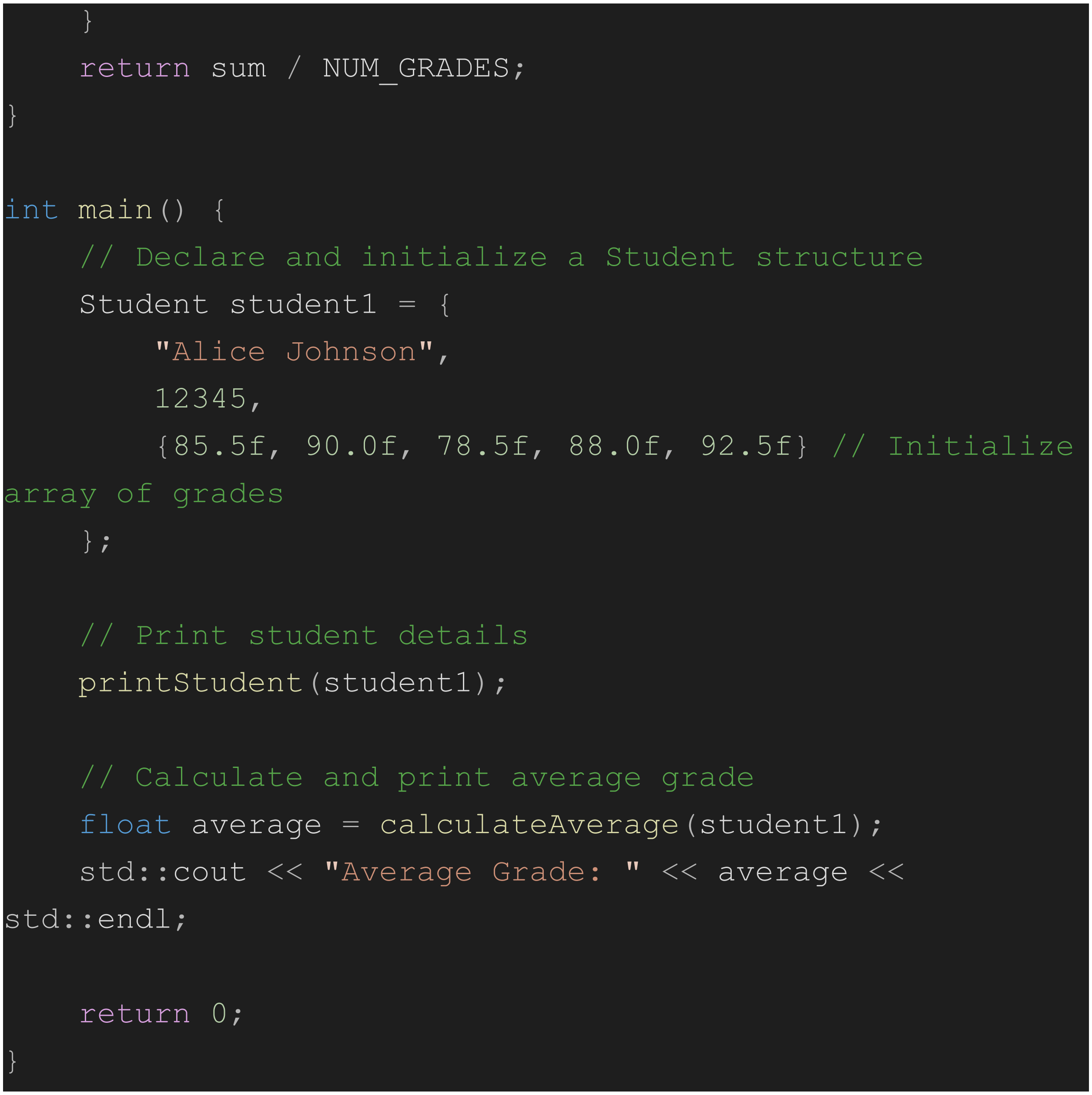
```
#include <iostream>
#include <string>

const int NUM_GRADES = 5; // Define the number of grades

// Define the Student structure
struct Student {
    std::string name;
    int id;
    float grades[NUM_GRADES]; // Array to hold grades
};

// Function to print student details
void printStudent(const Student &s) {
    std::cout << "Name: " << s.name << std::endl;
    std::cout << "ID: " << s.id << std::endl;
    std::cout << "Grades: ";
    for (int i = 0; i < NUM_GRADES; ++i) {
        std::cout << s.grades[i] << " ";
    }
    std::cout << std::endl;
}

// Function to calculate average grade
float calculateAverage(const Student &s) {
    float sum = 0.0f;
    for (int i = 0; i < NUM_GRADES; ++i) {
        sum += s.grades[i];
```



```
    }
    return sum / NUM_GRADES;
}

int main() {
    // Declare and initialize a Student structure
    Student student1 = {
        "Alice Johnson",
        12345,
        {85.5f, 90.0f, 78.5f, 88.0f, 92.5f} // Initialize
array of grades
    };

    // Print student details
    printStudent(student1);

    // Calculate and print average grade
    float average = calculateAverage(student1);
    std::cout << "Average Grade: " << average <<
std::endl;

    return 0;
}
```

National University
of computer and emerging sciences

Lab Tasks

Task 1

A Student identity card number can be thought of three parts: the campus code(I), the batch number (2022), and the ID (0392). Write a structure named as StudentCard to store these three parts of Student identity number. Data members should be named as campusCode of type char, batch of type int and ID of type int .

Task 2

Write a complete C++ program with the following features.

A. Declare a structure Point with two integer members x and y .

B. Define a function getInput(), which accepts a Point by reference. Get user input for a Point in this function.

C. Define a function addPoints(), which accepts two Points p1 and p2.

The function adds their respective members, and returns a Point which is the sum of two. For example if one point is (2,3), the other is (4,5), the function should return a Point (6,8). d. In the main(), declare two variables of type Point. C all the function getInput() twice to get the values of these Points from user. Add the two Points using the function addPoints() and display the x and y values of the result returned by the function.

Task 3

A. Declare a structure named as CustomTime having three data members named as ; hours of type int, min of type int and seconds of type int.

B. Write a function timeToSeconds which takes CustomTime object t1 as function parameter, Calculate the total seconds in time and convert this object t1 to seconds (of type long) and return it. long timeToSeconds(CustomTime t1)

C. Write a function SecondsToTime which takes long t as function parameter and return it in the form of a CustomTime object. CustomTime SecondsToTime(long t)

D. Write a function AddTimes which takes as input parameters CustomTime object t1 and CustomTime object t2 1. You have to convert these two objects t1, t2 in seconds using above defined function t timeToSeconds. 2. Add both seconds returned from above step. 3. Use above defined function SecondsToTime and convert back it to the CustomTime object and return it. CustomTime AddTimes(CustomTime t1, CustomTime t2)

Task 4

Declare a structure Rectangle with two Points as its members, the top left and the bottom right. i. Declare a variable o f type Rectangle and get user input for the two points. ii. Define a function computeArea() which accepts a Rectangle and returns its area. iii. Display the area of th e Rectangle in the main().