

Structs

Lecture 8

Outline

- Structures

Structures

- A data structure that can be used to store related data items with different types.
- The individual components of a struct is called a member.

Structures

Students

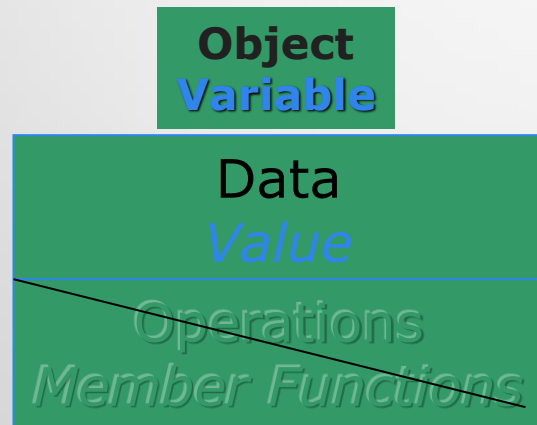
ID	Name	Major
1111	Nora	CS
2222	Sara	IS
3333	Mona	CS

- Student: ID variable
- Student: Name variable
- Student: Major variable

- ◆ Student
 - ID
 - Name
 - Major

Structures

Think of a structure as an object without any member functions



Here, we'll have values of different data types that we would like to treat as a *single item*.

Structures

➤ How do I....

- Define a structure?
- Use a structure?

■ Student

- ID
- Name
- Major

```
struct Student  
{  
    int    id;  
    char  name[10];  
    char  major[2];  
  
};
```

Structures

- **Syntax:**

```
struct Structure_Tag  
{  
    Type_1  Member_Variable_1;  
    Type_2  Member_Variable_2;  
  
    Type_n  Member_Variable_n;  
};
```

Structures

➤ Using Structures

- **Declare:**
StudentRecord Student1, Student2;
- **Assignment:** Student1 = Student2;
 - Student1.id = Student2.id;
 - Student1.grade = Student2.grade;
- **Read:** cin >> Student1.id;
- **Write:** cout << Student1.id;
- **Initialize:** Student1 = {666, 'A'}

Structures

- Syntax: Structure_Variable_Name ■ Member_Variable_Name

- Example:

```
struct StudentRecord
```

```
{
```

```
    int id;
```

```
    char grade;
```

```
};
```

```
int main ()
```

```
{
```

```
    StudentRecord    Student1;
```

```
    Student1.id      = 555;
```

```
    Student1.grade    = 'B';
```

```
    cout << Student1.id << ' ' << Student1.grade << endl;
```

```
}
```

Dot Operator



Structures

- ◆ Two or more structure types may use the same member names

```
struct FertilizerStock
{
    double quantity;
    double nitrogen_content;
};
```

```
struct CropYield
{
    int quantity;
    double size;
};
```

FertilizerStock Item1;



Item1.quantity

CropYield Apples;



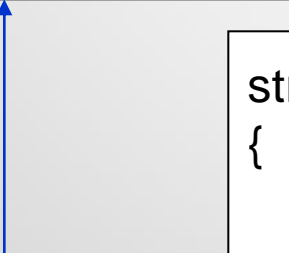
Apples.quantity

Structures

- Structures within structures (nested)

```
struct Date
{
    int month;
    int day;
    int year;
};
```

```
struct Employee
{
    int id;
    Date birthday;
};
```



```
Employee person1;
cout << person1.birthday.year;
```

Structures

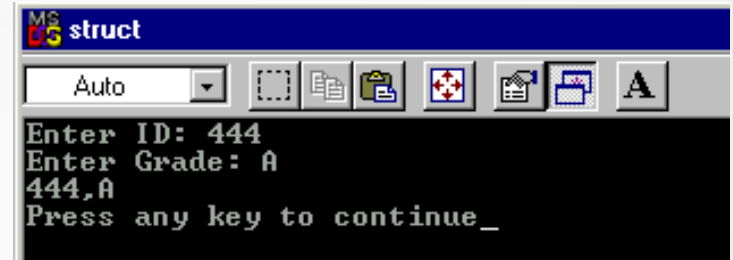
```
#include <iostream>

struct StudentRecord
{
    int id;
    char grade;
};

StudentRecord Get_Data (StudentRecord in_student);

int main ()
{
    using namespace std;
    StudentRecord Student1;
    Student1 = Get_Data (Student1);
    cout<< Student1.id<< ", "<<Student1.grade<< endl;
    return 0;
}

StudentRecord Get_Data (StudentRecord in_student)
{
    using namespace std;
    cout<<"Enter ID: "; cin>> in_student.id;
    cout<<"Enter Grade: "; cin>> in_student.grade;
    return (in_student);
}
```



Summary

- A struct is a data structure that has a collection of components and can be used as a data type.
- These components can be of different types and are called members.
- The keyword “struct” is used to create a data type.
- The “.” operator is used to access members of struct.
- A struct can contain another struct as its member.



Thank You