Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Structs

Comp Sci 1570 Introduction to C++

Missouri S&T | Computer Science

Outline

1 Introduction
General syntax
Object initialization
Initialization and access

2 Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()

3 Example
Single struct
Use in main()
Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

The struct

- A data structure is a group of data elements grouped together under one name.
- These data elements, known as members, can have different types and different lengths.
- Basic Object Oriented Programming (OOP)
- Allows us to abstract at a higher level to build entities more complex than short, long, int, float, double, char, and bool
- With it programmers can create their own types to define what should make up a student, a class, a department, a university, etc

Outline

1 Introduction
General syntax
Object initialization
Initialization and access
2 Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
3 Example
Single struct
Use in main()
Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

General syntax

```
struct type_name
{
    member_type1 member_name1;
    member_type2 member_name2;
                .
                .
                .
    member_typeN member_nameN;
};
```

- This code should be placed in a header file.
- If this new type has potential to be used in other programming projects, it should have its own header dedicated to its definition for ease of portability.
- In a lot of cases, the struct is particular to a project and it is fine to include it in the header file with all the function prototypes and global constants.

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

General syntax

```
struct type_name
{
    member_type1 member_name1;
    member_type2 member_name2;
                 .
                 .
                 .
    member_typeN member_nameN;
} object_names;
```

- Object names optional

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Initialization and access

```
struct type_name
{
    member_type1 member_name1;
};

int main(){
        type_name object_name;
        object_name.member_name1;
}
```

- To access any member of a structure, we use the member access operator '.'
- The member access operator is coded as a period between the structure variable name and the structure member that we wish to access.

Outline

1 Introduction
- General syntax
  - Object initialization
  - Initialization and access

2 Aggregate data
- Problems
- Solution: struct
- Declaration in main()
- Access in main()

3 Example
- Single struct
- Use in main()
- Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Aggregate data

- There are many instances in programming where we need more than one variable in order to represent an object.
- For example, to represent yourself, you might want to store your name, your birthday, your height, your weight, or any other number of characteristics about yourself.

```
string myName;
int myBirthYear;
int myBirthMonth;
int myBirthDay;
int myHeightInches;
int myWeightPounds;
```

Outline

1 Introduction
General syntax
Object initialization
Initialization and access

2 Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()

3 Example
Single struct
Use in main()
Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Aggregate data problems

- However, you now have 6 independent variables that are not grouped in any way.
- If you wanted to pass information about yourself to a function, you'd have to pass each variable individually.
- Furthermore, if you wanted to store information about someone else, you'd have to declare 6 more variables for each additional person!

```
string myName;
int myBirthYear;
int myBirthMonth;
int myBirthDay;
int myHeightInches;
int myWeightPounds;
```

Outline

1 Introduction
- General syntax
  - Object initialization
  - Initialization and access

2 Aggregate data
- Problems
- Solution: struct
- Declaration in main()
- Access in main()

3 Example
- Single struct
- Use in main()
- Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Aggregate data type: struct

- An aggregate data type is a data type that groups multiple individual variables together.
- One of the simplest aggregate data types is the struct.
- A **struct** (short for structure) allows us to group variables of mixed data types together into a single unit.
- Because structs are user-defined, we first have to tell the compiler what our struct looks like before we can begin using it.
- To do this, we declare our struct using the struct keyword. Here is an example of a struct declaration and definition:

```
struct Employee
{
    short id;
    int age;
    double wage;
};
```

Outline

1 Introduction
- General syntax
  - Object initialization
  - Initialization and access

2 Aggregate data
- Problems
- Solution: struct
- Declaration in main()
- Access in main()

3 Example
- Single struct
- Use in main()
- Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Declaration in main

In order to use the Employee struct, we simply declare variables of type Employee:

```
Employee joe; // create an Employee struct
Employee frank; // create another
```

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Outline

1 Introduction
General syntax
Object initialization
Initialization and access

2 Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()

3 Example
Single struct
Use in main()
Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Access in main

- When we define a variable such as Employee joe, joe refers to the entire struct (which contains the member variables).
- In order to access the individual members, we use the member selection operator (which is a period).
- Here is an example of using the member selection operator to initialize each member variable:

```
Employee joe; //create an Employee struct
joe.id=14; //assign value to member id in joe
joe.age=32; //assign value to member age
joe.wage=24.15;

Employee frank;
frank.id=15;
frank.age=28;
frank.wage=18.27;
```

Outline

1 Introduction
General syntax
Object initialization
Initialization and access

2 Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()

3 Example
Single struct
Use in main()
Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Outline

1 Introduction
- General syntax
  - Object initialization
  - Initialization and access

2 Aggregate data
- Problems
- Solution: struct
- Declaration in main()
- Access in main()

3 Example
- Single struct
- Use in main()
- Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

A simple struct

```
// this code to be placed in a header file
struct point
{
    float m_Xcoord;
    float m_Ycoord;
};
```

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Outline

1 Introduction
  General syntax
    Object initialization
    Initialization and access
2 Aggregate data
  Problems
  Solution: struct
  Declaration in main()
  Access in main()
3 Example
  Single struct
  Use in main()
  Struct of structs

In main()

```
int main()
{
  point p1, p2;
  P1.m_Xcoord = 4;
  P1.m_Ycoord = 6;
  cout << "enter␣p2's␣x:␣";
  cin >> p2.m_Xcoord;
  cout << "and␣the␣y:␣";
  cin >> p2.m_Ycoord;
  cout << "the␣x␣coordinate␣of␣p1␣is␣"
       << p1.m_Xcoord;
...
```

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Outline

1 Introduction
- General syntax
  - Object initialization
  - Initialization and access

2 Aggregate data
- Problems
- Solution: struct
- Declaration in main()
- Access in main()

3 Example
- Single struct
- Use in main()
- Struct of structs

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Struct of structs

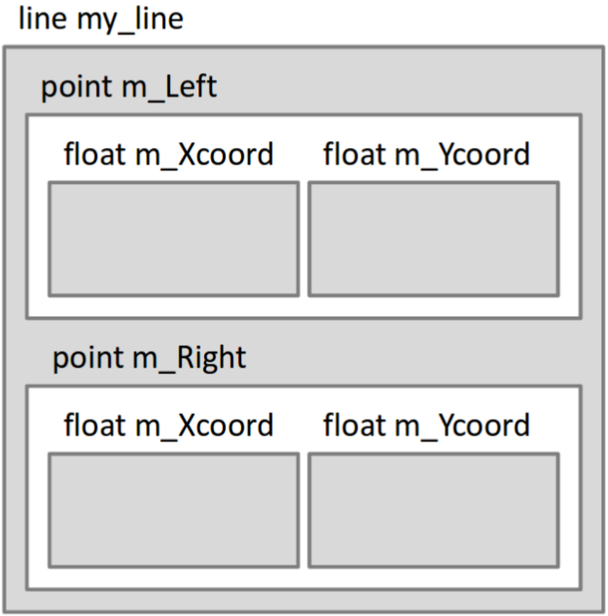
```
struct point
{
    float m_Xcoord;
    float m_Ycoord;
};

struct line
{
    point m_Left;
    point m_Right;
};

int main()
{
    line my_line; // line object
    my_line.m_Left.m_Xcoord = 5; // point obj.
    my_line.m_Left.m_Ycoord = 8; // ...
```

Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Struct of structs



Introduction
General syntax
Object initialization
Initialization and access
Aggregate data
Problems
Solution: struct
Declaration in main()
Access in main()
Example
Single struct
Use in main()
Struct of structs

Carpart struct example

```
struct carpart
{
    string m_description;
    long m_partNumber;
    float m_wholesalePrice;
    float m_retailPrice;
    string m_color;
    etc
};
```