Lecture-18

typedef, Aggregate datatypes: *enum*, *struct*, *union* and *class*

typedef

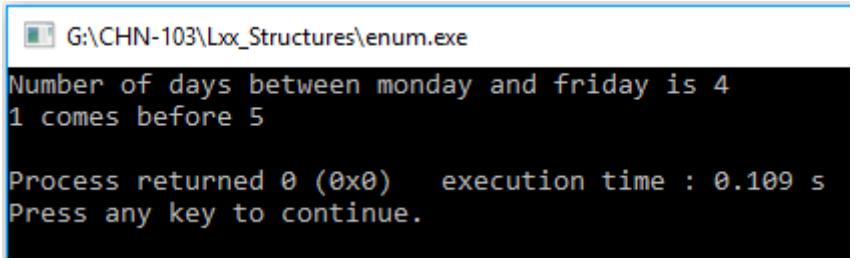
Used for creating synonyms (aliases) for already existing data types.

- Shorter, simpler or more readable names can be defined
- Can be used to make programs portable.

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

typedef int integer;
typedef float real;

typedef char name[20];
typedef char school[50];

int main(){

    integer n1 = 4, n2 = -2;
    real r1 = 3.56, r2 = -4.32;

    cout << "The sum of integers is " << (n1 + n2) << endl;
    cout << "The product of real numbers is " << (r1*r2) << endl;

    name studentName = "raja";
    school schoolName = "abc driving institute";

    cout << studentName << " is learning at " << schoolName << endl;

}
```

```
G:\CHN-103\Lxx_Structures\typedef.exe
The sum of integers is 2
The product of real numbers is -15.3792
raja is learning at abc driving institute

Process returned 0 (0x0)   execution time : 0.125 s
Press any key to continue.
```

enum datatype



Enumeration is assigning numeric values to non-numeric data

- Allows for operators to be used on non-numeric data as if they were numbers.

```
1  #include <iostream>
2  using namespace std;
3
4  enum days{
5      sunday, monday, tuesday, wednesday, thursday, friday, saturday
6  };
7
8  int main(){
9
10     days d1, d2;
11     d1 = monday;
12     d2 = friday;
13
14     cout << "Number of days between monday and friday is " << abs(d1 - d2) << endl;
15
16     if (d1 < d2)
17         cout << d1 << " comes before " << d2 << endl;
18     else
19         cout << d2 << " comes before " << d1 << endl;
20
21 }
```

Note: numbering starts with 0 by default, and can be changed by assigning different number.

struct datatype

- Ported from c language, c++ uses *struct* datatype just like classes, for combining different data types and member functions.
- In *struct* all members are *public* by default, while in *class* all members are *private* by default.

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

typedef char name[20];
typedef unsigned short ushrt;

struct student{
    name            student_name;
    ushrt           age;
    float           height;
    float           weight;
    float           marks;
}s1,s2;

int main(){
    cout << "Enter the name of student: ";
    cin.getline(s1.student_name,20);
    cout << s1.student_name << endl;

    student s3 = {"vishal",20,1.56,68.5,78};

    cout << "Our student is: " << endl;
    cout << "Student name: " << s3.student_name << endl
         << "Age: " << s3.age << endl
         << "Height: " << s3.height << endl
         << "Weight: " << s3.weight << endl
         << "Marks obtained: " << s3.marks << endl;
}
```

```
G:\CHN-103\Lxx_Structures\structStudent.exe
Enter the name of student: Raju
Raju
Our student is:
Student name: vishal
Age: 20
Height: 1.56
Weight: 68.5
Marks obtained: 78

Process returned 0 (0x0)   execution time : 4.172 s
Press any key to continue.
```

struct datatype

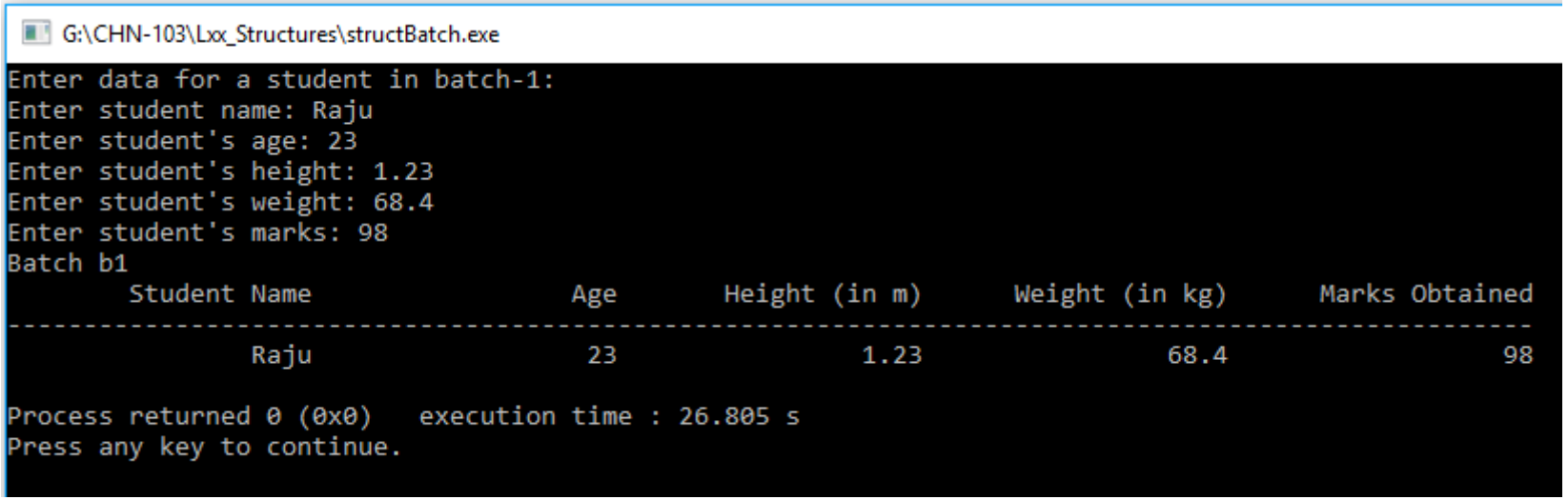
```
1  #include <iostream>
2  #include <iomanip>
3  #define WID 20
4  using namespace std;
5  typedef char name[20];
6  typedef unsigned short ushrt;
7
8  struct student{
9      name            student_name;
10     ushrt           age;
11     float           height;
12     float           weight;
13     float           marks;
14 }s1,s2;
15
16 struct batch{
17
18     student     s[10];
19     float       avg_marks;
20     float       highest_marks;
21     float       lowest_marks;
22
23 }b1,b2;
```

struct batch uses *struct student* to create nested data structure

struct datatype

```
25  int main(){
26
27      cout << "Enter data for a student in batch-1: " << endl;
28      cout << "Enter student name: "; cin >> b1.s[0].student_name;
29      cout << "Enter student's age: "; cin >> b1.s[0].age;
30      cout << "Enter student's height: "; cin >> b1.s[0].height;
31      cout << "Enter student's weight: "; cin >> b1.s[0].weight;
32      cout << "Enter student's marks: "; cin >> b1.s[0].marks;
33
34      // Display batch data
35      cout << "Batch b1" << endl;
36      cout << setw(WID) << "Student Name"
37           << setw(WID) << "Age"
38           << setw(WID) << "Height (in m)"
39           << setw(WID) << "Weight (in kg)"
40           << setw(WID) << "Marks Obtained" << endl;
41
42      cout << setw(WID) << "--------------------"
43           << setw(WID) << "--------------------"
44           << setw(WID) << "--------------------"
45           << setw(WID) << "--------------------"
46           << setw(WID) << "--------------------" << endl;
47
48      cout << setw(WID) << b1.s[0].student_name
49           << setw(WID) << b1.s[0].age
50           << setw(WID) << b1.s[0].height
51           << setw(WID) << b1.s[0].weight
52           << setw(WID) << b1.s[0].marks << endl;
53  }
```

struct datatype



struct datatype

```
1   #include <iostream>
2   #include <iomanip>
3   #define WID 20
4   using namespace std;
5   typedef char name[20];
6   typedef unsigned short ushrt;
7
8   struct student{
9
10      name            student_name;
11      ushrt           age;
12      float           height;
13      float           weight;
14      float           marks;
15
16  }s1,s2;
17
18  struct batch{
19
20      short       num_students;
21      student     *s;
22      float       avg_marks;
23      float       highest_marks;
24      float       lowest_marks;
25
26  }b1,b2;
```

struct batch uses pointer to *struct student* to create nested data structure

struct datatype

```
28  void getData(batch);
29  void display(batch);
30  float average(batch);
31  float highest(batch);
32  float lowest(batch);
33
34  int main(){
35
36      cout << "Enter the number of students in batch: ";
37      cin >> b1.num_students;
38      b1.s = new student[b1.num_students];
39      getData(b1);
40      display(b1);
41      cout << "Average marks for batch is: " << average(b1) << endl;
42  }
43
44  void getData(batch b){
45      for (int i = 0; i < b.num_students; i++){
46          cout << "Enter student [" << i << "] name: "; cin >> (b.s+i)->student_name;
47          cout << "Enter student [" << i << "] age: "; cin >> (b.s+i)->age;
48          cout << "Enter student [" << i << "] height: "; cin >> (b.s+i)->height;
49          cout << "Enter student [" << i << "] weight: "; cin >> (b.s+i)->weight;
50          cout << "Enter student [" << i << "] marks: "; cin >> (b.s+i)->marks;
51      }
52  }
```

Allocate memory for batch as per user requirement

struct datatype

```
54  void display(batch b){
55
56      cout << setw(WID) << "Student Name"
57           << setw(WID) << "Age"
58           << setw(WID) << "Height (in m)"
59           << setw(WID) << "Weight (in kg)"
60           << setw(WID) << "Marks Obtained" << endl;
61
62      cout << setw(WID) << "--------------------"
63           << setw(WID) << "--------------------"
64           << setw(WID) << "--------------------"
65           << setw(WID) << "--------------------"
66           << setw(WID) << "--------------------" << endl;
67
68      for (int i = 0; i < b.num_students; i++){
69          cout << setw(WID) << (b.s+i)->student_name
70               << setw(WID) << (b.s+i)->age
71               << setw(WID) << (b.s+i)->height
72               << setw(WID) << (b.s+i)->weight
73               << setw(WID) << (b.s+i)->marks << endl;
74      }
```

struct datatype

```
78  float average(batch b){
79
80      float sum = 0;
81      for (int i = 0; i < b.num_students; i++){
82          sum += (b.s+i)->marks;
83      }
84
85      b.avg_marks = sum/(b.num_students);
86      return (b.avg_marks);
87
88  }
```

Implement functions for finding highest and lowest marks.

struct datatype

G:\CHN-103\Lxx_Structures\structPointer.exe

```
Enter the number of students in batch: 5
Enter student [0] name: Raju
Enter student [0] age: 23
Enter student [0] height: 1.23
Enter student [0] weight: 65.4
Enter student [0] marks: 87
Enter student [1] name: Sadiq
Enter student [1] age: 24
Enter student [1] height: 1.24
Enter student [1] weight: 78
Enter student [1] marks: 84
Enter student [2] name: Reema
Enter student [2] age: 22
Enter student [2] height: 1.3
Enter student [2] weight: 60
Enter student [2] marks: 86
Enter student [3] name: Perez
Enter student [3] age: 22
Enter student [3] height: 1.26
Enter student [3] weight: 62
Enter student [3] marks: 78
Enter student [4] name: Shiva
Enter student [4] age: 23
Enter student [4] height: 1.23
Enter student [4] weight: 67.4
Enter student [4] marks: 84
        Student Name                 Age       Height (in m)      Weight (in kg)      Marks Obtained
-------------------------------------------------------------------------------------------------
                Raju                  23                1.23                65.4                  87
               Sadiq                  24                1.24                  78                  84
               Reema                  22                 1.3                  60                  86
               Perez                  22                1.26                  62                  78
               Shiva                  23                1.23                67.4                  84
Average marks for batch is: 83.8

Process returned 0 (0x0)   execution time : 125.252 s
Press any key to continue.
```