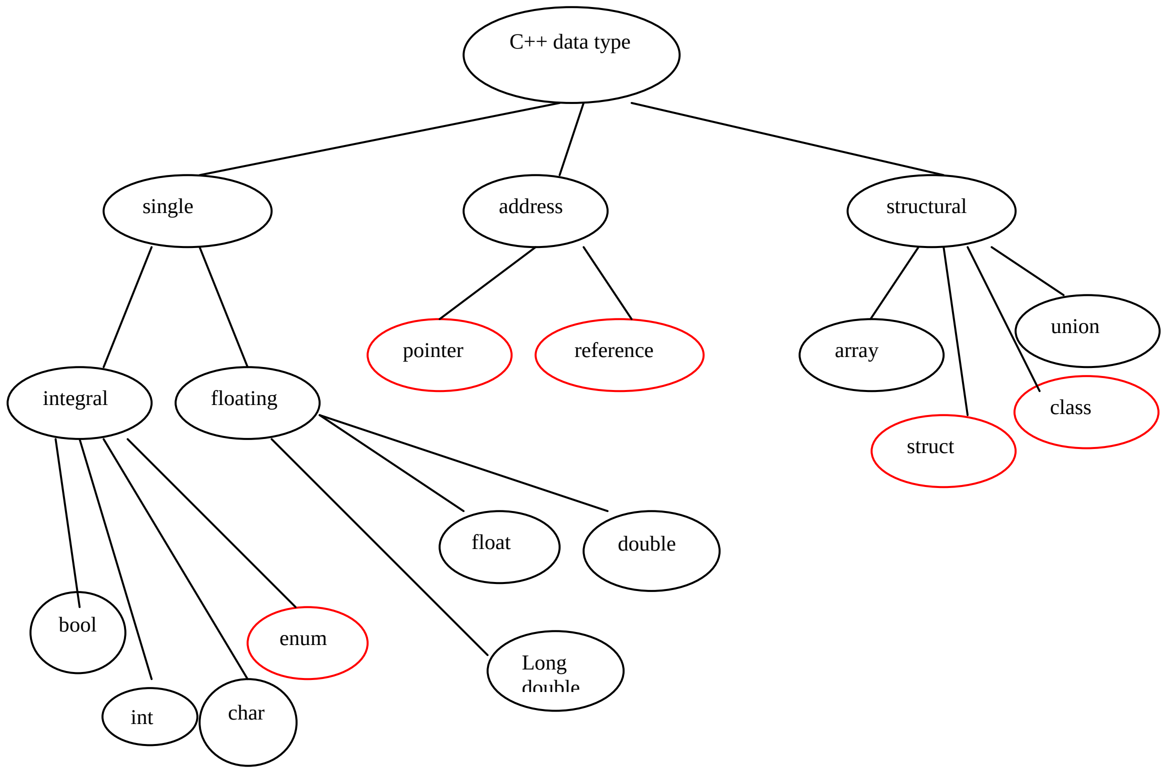
CSCI 2170 **Lecture notes (2)**
enum type

- **C++ data type**



Signed
Unsigned
Long
short

- **enumeration type : a user defined data type whose domain is an ordered set of literal values expressed as identifiers**
    - o enhance program readability
    - o identifiers have to be unique, values do not have to be unique

syntax:

```
enum enumerative-type {enumerator list};
```

Example:

```
enum    day  {SUN, MON, TUE, WED, THU, FRI, SAT};
// default value of the first identifier is 0, and every subsequent identifier have a value
that is 1 greater than its previous identifier
// Enumerators are like named constants. It is equivalent to:

const int SUN = 0;
const int MON = 1;
```

```
…
const int SAT = 6;
```

The difference is that, with enum, now we can create variables of this new type. The value of the variables are limited to the one defined for this type.

```
// the internal value of the enumerators can be changed:
enum day {SUN=4, MON=10, TUE=8, …};
     ------  identifier rules apply, the following is wrong:
                    enum day {'S', 'M', 'T', 'W', …};
```

- **create variable of enum type**

```
day    birthday;
```

- **assignment**

```
birthday = TUE;
// wrong:    birthday = 2;
```

- **comparison**

```
…
cout << "Your birthday is on ";
switch (birthday)
{
        case SUN:      cout << "Sunday." << endl;
                               break;
        case MON:      cout << "Monday." << endl;
                               break;
        case TUE:      cout << "Tuesday." << endl;
                               break;
        …
}
```

Example: enum coin {PENNY, NICKEL, DIME, QUARTER, DOLLAR};

```
coin money;
money = DIME;                              // assignment
cout << money;                             // output

cout << PENNY << '\t' << NICKEL << endl;

if (money == QUARTER)                      // comparison
    cout << "Got a quarter";
else
    cout << "not a quarter";
```

- **incrementation**

// wrong : **// correct**

```
// wrong  :                          // correct
money = money +1;    or              money = coin(money+1);
money ++;
```

- **input**

```
// Not allowed: cin >> money;           // extraction operator does not work with enum type
// then how?

#include <cctype>
#include<string>

enum Animals {RODENT, CAT, DOG, BIRD, REPTILE, HORSE, SHEEP};
Animal inPatient;
string animalName;

cin >> animalName;
switch (toupper(animalName[0]))
{
    case ‘R’ :   if (toupper(animalName[1]) == ‘O’)
                         inPatient = RODENT;
                 else
                         inPatient = REPTILE;
                  break;
    case ‘C’ :   inPatient = CAT;
                 break;
    case ‘D’ :   inPatient = DOG;
                 break;
     case ‘B’ :  inpatient = BIRD;
                 break;
     case ‘H’ :  inpatient = HORSE;
                 break;
     default:    inpatient = SHEEP;
}
```

- **use enum type as array index**

```
const int SIZE = 6;
int    count [SIZE];

coin money;
for (money=PENNY; money<=DOLLAR; money = coin(money+1))
        count[money] = 0;
```