CSCI 2170 enum type

- **enumeration type : a user defined data type whose domain is an ordered set of literal values expressed as identifiers**
    - enhance program readability
    - 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
// Naming enumerators follows the same rules for naming identifiers
//  enum day {'S', 'M', 'T', 'W', …};    is incorrect

// Enumerators are like named constants. It is equivalent to:
      const int SUN = 0;                          …
      const int MON = 1;                          const int SAT = 6;
```

- **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**
```
// incorrect                      // correct
money = money +1;                 money = coin(money+1);
money ++;
```

- **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;
```

- **change enumerator value**

```
// the internal value of the enumerators can be changed:
enum day {SUN=4, MON=10, TUE=8, …};
```

- **input**
  - Not allowed:

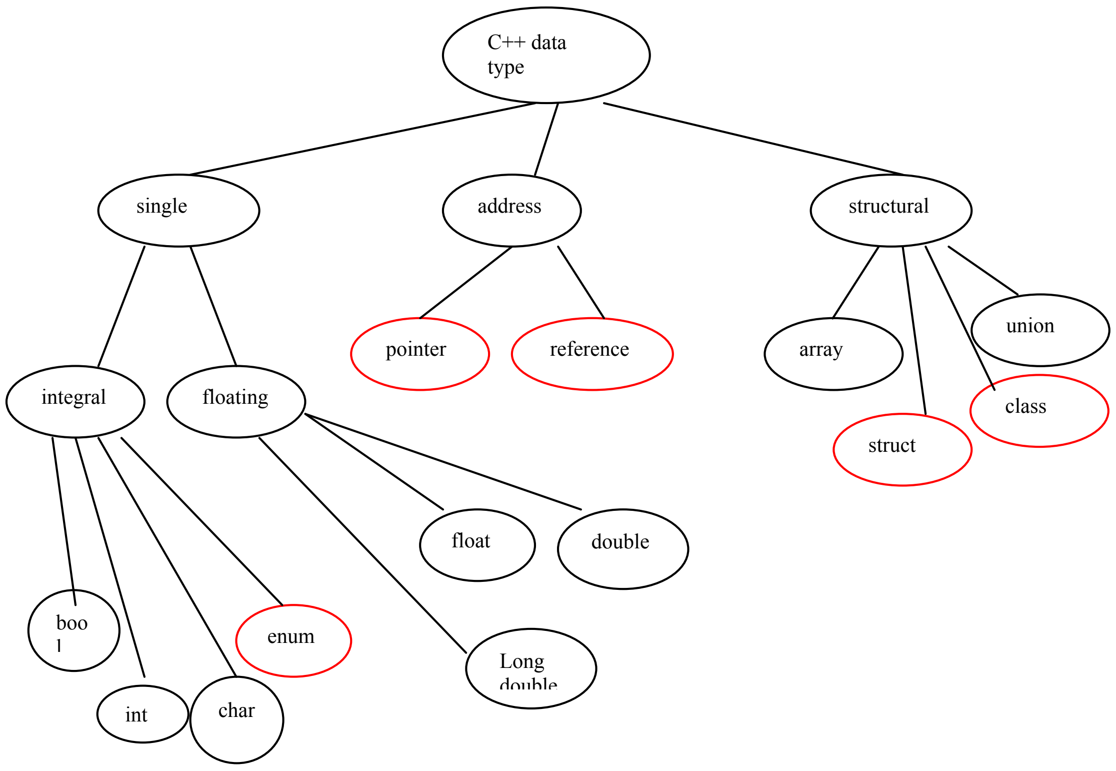
    ```
    cin >> money;          // extraction operator does not work with enum type
    ```

  - **Correct example**

    ```
    #include <cctype>

    enum Animals {RODENT, CAT, DOG, BIRD, REPTILE, HORSE, SHEEP};
    Animal inPatient;
    char animalName[25];

    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;
    }
    ```



Signed
Unsigned
Long
short

C++ data type