

# C++ Programming

# เปรียบเทียบภาษา

## ตัวอย่างโปรแกรมภาษา C++

```
1 //read 100 numbers from the user and
2 //output their sum
3
4 #include <iostream>
5 using namespace std;
6
7 int main() {
8     int i, sum, x;
9     sum=0;
10    i=1;
11    while(i <=100) {
12        cin >> x;
13        sum = sum+x;
14        i=i+1;
15    }
16    cout << "sum is" << sum << endl;
17 }
```

## ตัวอย่างโปรแกรมภาษา C

```
1 //read 100 numbers from the user and
2 //output their sum
3
4 #include <stdio.h>
5
6
7 int main() {
8     int i, sum, x;
9     sum=0;
10    i=1;
11    while(i <=100) {
12        scanf("%d", &x);
13        sum = sum+x;
14        i=i+1;
15    }
16    printf("sum is %d \n",sum);
17 }
```

### ตัวอย่างโปรแกรมภาษา C++

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main () {
6     string str;
7     str = "The initial";
8     cout << str << endl;
9     str = "THE different";
10    cout << str << endl;
11    return 0;
12 }
```

### ตัวอย่างโปรแกรมภาษา C

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main () {
5     char str[12] = "The initial";
6     char ss[14] = "THE different";
7     printf("%s \n", str);
8     strcpy(str,ss);
9     printf("%s \n", str);
10    return 0;
11 }
```

# Using standard classes

## Sample program using class string

```
// To use strings.

#include <iostream>    // Declaration of cin, cout
#include <string>      // Declaration of class string
using namespace std;

int main()
{
    // Defines four strings:
    string prompt("What is your name: "),
           name, // An empty
           line( 40, '-'), // string with 40 '-'
           total = "Hello "; // is possible!

    cout << prompt; // Request for input.
    getline( cin, name); // Inputs a name in one line

    total = total + name; // Concatenates and
                        // assigns strings.

    cout << line << endl // Outputs line and name
         << total << endl;
    cout << " Your name is " // Outputs length
         << name.length() << " characters long!" << endl;
    cout << line << endl;
    return 0;
}
```

## Creating object

- An object is a variable of a class type
- referred to as an instance of the class
- ยกตัวอย่าง

```
string s("I am a string");
```

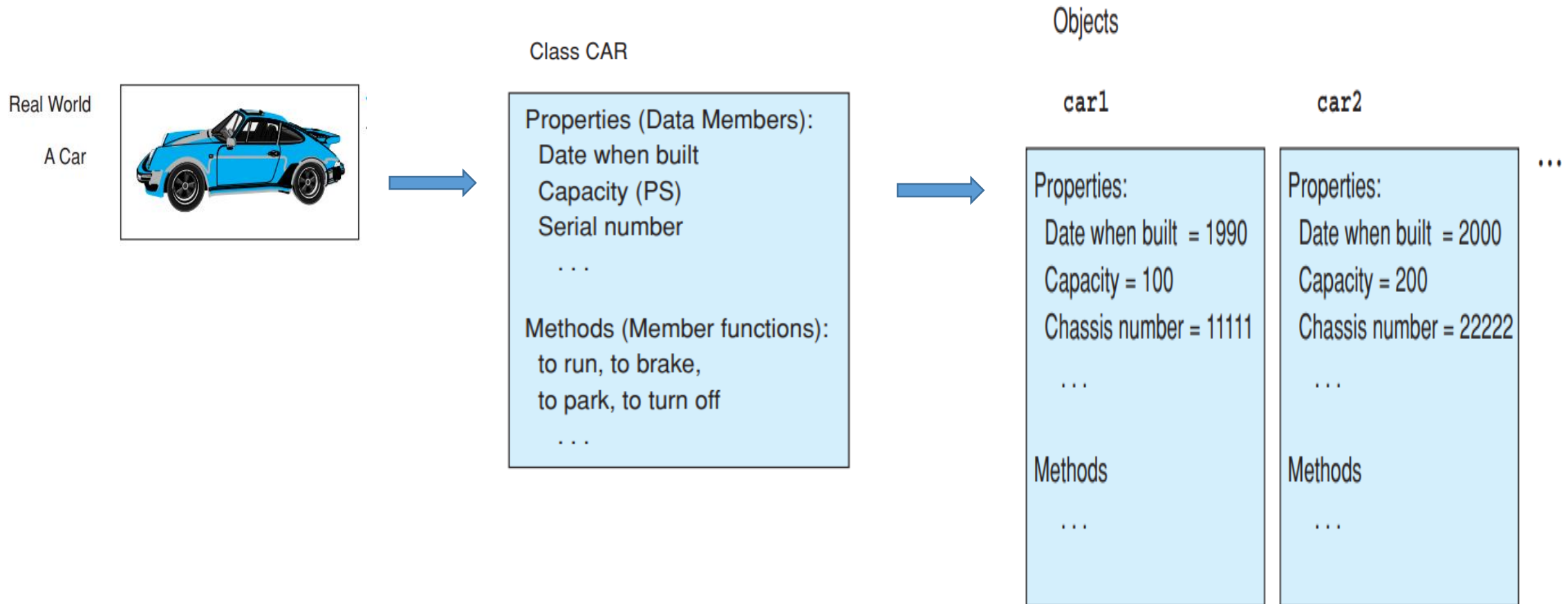
## Calling methods

- All the methods defined as **public** within the corresponding **class** can be called for an **object**
- ยกตัวอย่าง

```
s.length(); // object.method();
```

# Defining classes

- Class concept



# Example of class

```
// account.h
// Defining the class Account.
// -----
#ifndef _ACCOUNT_          // Avoid multiple inclusions.
#define _ACCOUNT_

#include <iostream>
#include <string>
using namespace std;

class Account
{
    private:                // Sheltered members:
        string name;        // Account holder
        unsigned long nr;   // Account number
        double balance;     // Account balance

    public:                 //Public interface:
        bool init( const string&, unsigned long, double);
        void display();
};

#endif    // _ACCOUNT_
```

# Methods of class account

```
// account.cpp
// Defines methods init() and display().
// -----
#include "account.h" // Class definition
#include <iostream>
#include <iomanip>
using namespace std;

// The method init() copies the given arguments
// into the private members of the class.
bool Account::init(const string& i_name,
                  unsigned long i_nr,
                  double i_balance)
{
    if( i_name.size() < 1) // No empty name
        return false;
    name = i_name;
    nr = i_nr;
    balance = i_balance;
    return true;
}

// The method display() outputs private data.
void Account::display()
{
    cout << fixed << setprecision(2)
         << "-----\n"
         << "Account holder:   " << name << '\n'
         << "Account number:    " << nr << '\n'
         << "Account balance:    " << balance << '\n'
         << "-----\n"
         << endl;
}
```



# Using objects

```
// account_t.cpp
// Uses objects of class Account.
// -----

#include "Account.h"

int main()
{
    Account current1, current2;

    current1.init("Cheers, Mary", 1234567, -1200.99);
    current1.display();

    //    current1.balance += 100; // Error: private member

    current2 = current1;           // ok: Assignment of
                                   // objects is possible.
    current2.display();           // ok

                                   // New values for current2
    current2.init("Jones, Tom", 3512347, 199.40);

    current2.display();

    Account& mtr = current1;       // To use a reference:
                                   // mtr is an alias name
                                   // for object current1.
    mtr.display();                // mtr can be used just
                                   // as object current1.

    return 0;
}
```



# ตัวอย่างการใช้ array

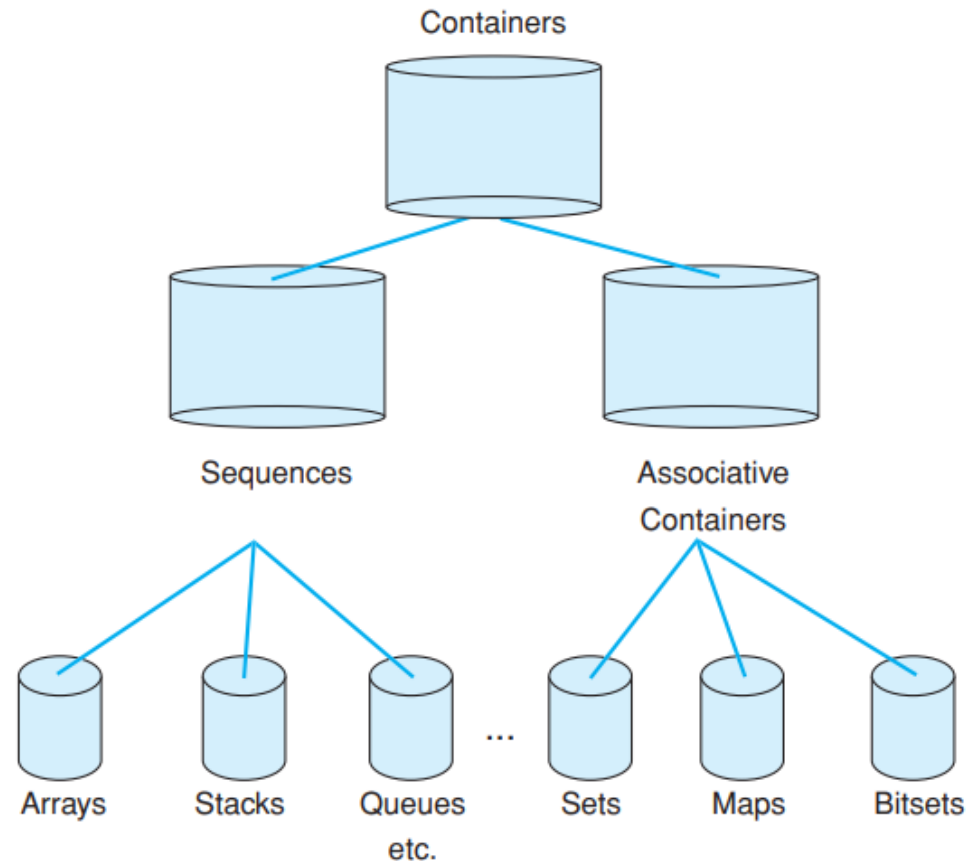
```
// array.cpp
// To input numbers into an array and output after.
// -----
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    const int MAXCNT = 10;           // Constant
    float arr[MAXCNT], x;           // Array, temp. variable
    int i, cnt;                     // Index, quantity

    cout << "Enter up to 10 numbers \n"
         << "(Quit with a letter):" << endl;
    for( i = 0; i < MAXCNT && cin >> x; ++i)
        arr[i] = x;
    cnt = i;
    cout << "The given numbers:\n" << endl;
    for( i = 0; i < cnt; ++i)
        cout << setw(10) << arr[i];
    cout << endl;
    return 0;
}
```

# Containers

- Are used to store objects of the same type
- Provide operations with which these objects can be managed
- For more efficient management of object collections



# Set

- are used for efficient management of object collections with sortable keys.
- insertion, deletion, and search operations can be performed with logarithmic runtimes
- insert() method is available for insertions.
- erase() method to delete objects.
- find() method to search objects.

ที่มา <https://cplusplus.com/reference/set/set/>

# ตัวอย่างโปรแกรม set

```
1  // set_t.cpp
2  #include <set>
3  #include <cstdlib>
4  #include <ctime>
5  #include <iostream>
6  using namespace std;
7
8  typedef set<int> IntSet;          //Define typedef
9  typedef IntSet::iterator SetIter; //Define iterator typedef
10
11 int main() {
12     IntSet lotto;          //Create a set
13     SetIter pos;          //Bidirectional iterator
14
15     srand((unsigned) time(NULL));
16     while(lotto.size() < 6) //Insert
17         lotto.insert(rand() % 50);
18
19     cout << "These are your lotto numbers: " << endl;
20     for(pos = lotto.begin(); pos != lotto.end(); pos++)
21         cout << *pos << " ";
22     cout << endl << endl;
23
24     return 0;
25 }
```

# ตัวอย่างโปรแกรม set

```
1  #include <set>
2  #include <iostream>
3  using namespace std;
4
5  typedef set<int> IntSet;           //Define typedef
6  typedef IntSet::iterator SetIter; //Define iterator typedef
7
8  int main() {
9      IntSet mySet;           //Create a set
10     SetIter pos;           //Bidirectional iterator
11
12     for(int i=1; i<=5; i++) //Insert
13         mySet.insert(i*10); // set: 10 20 30 40 50
14
15     pos = mySet.find(20);
16     mySet.erase(pos);
17
18     cout << "MySet contains : " << endl;
19     for(pos = mySet.begin(); pos != mySet.end(); pos++)
20         cout << *pos << " ";
21     cout << endl << endl;
22
23
24     return 0;
25 }
```

# iterator(ตัววนซ้ำ)

- เป็นตัวชี้(pointer) แบบละเอียด
- The begin() method accesses the first position
- The end() method accesses after the last container object

# References

- A complete guide to programming in C++, Ulla Kirch-Prinz.
- <http://www.lmpt.univ-tours.fr/~volkov/C++.pdf>