BAKI ALİ NEFT MƏKTƏBİ
BAKU HIGHER OIL SCHOOL

Programming and Computer Applications

Classes
A Deeper Look

Instructor : PhD, Associate Professor Leyla Muradkhanli

Outline

- Introduction
- `const` Objects and `const` Member Functions
- Composition: Objects as Members of Classes
- `friend` Functions and `friend` Classes
- Using the `this` Pointer
- `static` Class Members

Introduction

- **`const` objects and `const` member functions**
  - prevent modifications of objects and enforce the principle of least privilege.
- **Composition**
  - a form of reuse in which a class can have objects of other classes as members.
- **Friendship**
  - enables a class designer to specify nonmember functions that can access a class's non-`public` members
- **The `this` pointer**
  - an implicit argument to each of a class's non-`static` member functions.
  - allows those member functions to access the correct object's data members and other non-`static` member functions.

`const` Objects and `const` Member Functions

- You may use keyword `const` to specify that an object is not modifiable and that any attempt to modify the object should result in a compilation error.
- A member function is specified as `const` *both in its prototype and in its definition.*

Composition: Objects as Members of Classes

- Composition
  - Sometimes referred to as a has-a relationship
  - A class can have objects of other classes as members
- An object's constructor can pass arguments to member-object constructors via member initializers.

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

class Date
{
private:
int day;
int month;
int year;

public:
Date(int=1, int=1, int=2010);
void print();
};
```

```
class Employee
{
public:
    Employee(string, string, Date, Date);
    void print();

private:
    string firstName;
    string lastName;
    Date birthDate;
    Date hireDate;
};
```

```
Date::Date(int dy, int mn, int yr)
{
day=dy;
month=mn;
year=yr;
}

void Date::print()
{
cout<<day<<'/'<<month<<'/'<<year;
}
```

```
Employee::Employee(string first, string last,
    Date dateofbirth, Date dateofhire)
{
 firstName =first;
 lastName =last;
 birthDate=dateofbirth;
 hireDate=dateofhire;
}
void Employee::print()
{
cout<<firstName<<' '<<lastName<<" Hired : ";
hireDate.print();
cout<<" Birthday : ";
birthDate.print();
cout<<endl;
}
```

```
int main()
{
Date birth(27,5,2001);
Date hire(3,12,2020);
Employee manager("Ali", "Aliyev",birth, hire);
cout<<endl;
manager.print();
cout<<"\nTest Date constructor with default values :\n";
Date birth2;
Date hire2;
Employee manager2("Anar", "Mammadov", birth2, hire2);
cout<<endl;
manager2.print();
return 0;
}
```