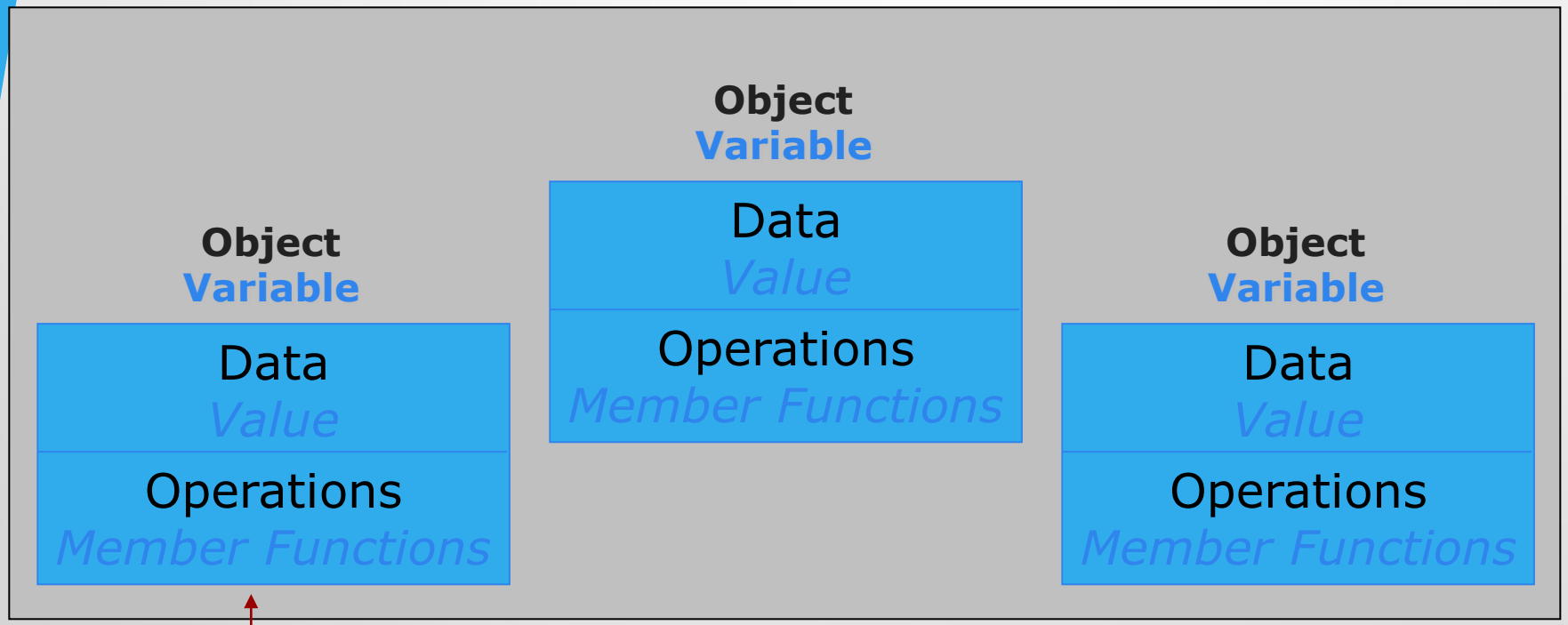
Classes

Lecture 9

Outline

- Classes
- Classes & ADTs
- 6.2 Classes

Classes



Encapsulation: combining a number of items such as variables and functions into a single package (object).

Classes

Syntax:

```
class Class_Name
{
public:
    Member_Specification_1
    Member_Specification_2
    ....
    Member_Specification_n
private:
    Member_Specification_n+1
    Member_Specification_n+2
    ....
};
```

public members

private members

Classes

Example:

```
class Bicycle
{
public:
        char  get_color();
        int   number_of_speeds();
        void set (int the_speeds, char the_color);
private:
        int speeds;
        char color;
};
```

public members

private members

```
Bicycle  my_bike, your_bike
```

Classes

Member Function Syntax:

```
Return_Type   Class_Name :: Function_Name (Parameter_List)
{
    Function_Body_Statements
};
```

```
class DayOfYear
{
public:
      void      output();
      int   month;
      int   day;
};
```

Example:

```
void DayOfYear::output()
{
    cout<< "month= "<<month<<"day= "<<day<<endl;
}
```

Classes

- *Use the dot (.) operator to access the members*

```
DayOfYear  today;

today.month = 2;
today.day = 10;

today.output();
```

```
class DayOfYear
{
public:
     void    output();
     int  month;
     int  day;
};
```

Classes

Dot Operator (.)

- Used with Objects – class variables
- Example:

```
today.output();
```

Scope Resolution Operator (::)

- Used with Class name
- Example:

```
void DayOfYear::output()
```

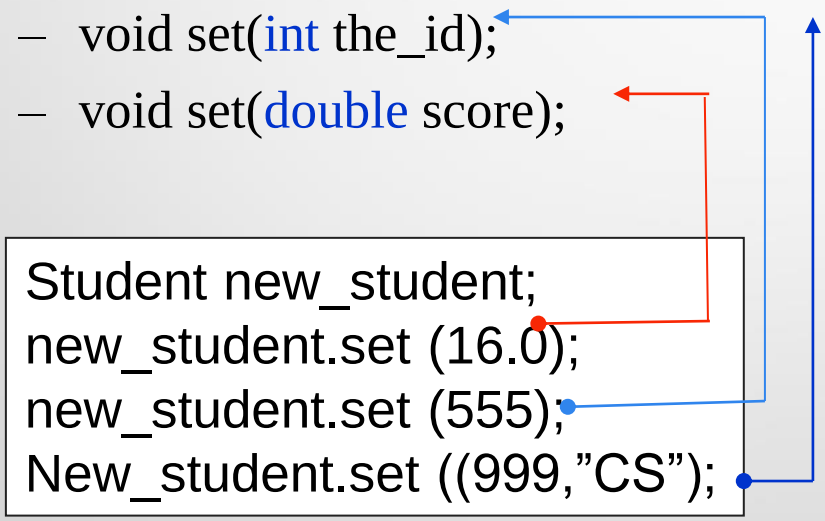
Classes & ADTs

Public Vs Private

- Separate the *rules for using* the class and the details of the class *implementation*
- Have enough member functions that you never need to access member variables directly, only through member functions
- Code is easier to *understand & update*

Classes & ADTs

- Can we overload member functions?
  - void set(int the_id, char the_major[2]);
  - void set(int the_id);
  - void set(double score);



Search for matching data types and/or number of parameters

6.2 Classes

Member Function Definition

```
void   DayOfYear::output()
{
      cout<< “month= “<< month;
      cout<<“day=“ << day;
      cout<<endl;
}
```

```
class DayOfYear
{
public:
      void      output();
private:
      int   month;
      int   day;
};
```

Private members may be used in member function definitions (but not elsewhere).

6.2 Classes

```
class Sample
{
public:
      int   variable;
      void      output();
      void      input();
private:
      int   month;
      int   day;
      void      doStuff();
};
```

public members

private members

Public members can be used in the main body of your program or in the definition of any function, even a non- member function.

Summary

- A class is a data structure that has member attributes and methods.
- The keyword “class” is used to create a new class.
- Overloaded methods have the same name but different parameters.
- The “.” operator is used to access members of a class.
- Private members are only visible within the class.
- Public members can be used from anywhere outside the class.

Thank You