Programming Technique II – SCJ1023

Classes and Object Manipulation

Associate Prof. Dr. Norazah Yusof

What is instance members?

- Instance members are member variables or member functions in a class, where the variables in an object are separated and distinct from the member variables of other objects of the same class.
- Each object had it has its own copy of instance variables.
- An instance member function can be used to access instance member variable of the class.

What is a static members?

- Static members are member variables or member function that does not belong to any instance of a class.
- A static variable share data among all objects of a class.
- A static member function is used to access static member variables.

Example of instance dan static members

```
class acStaff{
      int dept;
      static int faculty;
  public:
    acStaff()
    {dept=0;}
    acStaff(int d)
    {dept=d; addFaculty(d);}
    void addDept(int num)
    { dept+=num;}
    static void addFaculty(int num)
    { faculty+=num;}
    int getDept() const
    { return dept;}
    int getfaculty() const
    { return faculty;}
};
```

What is a friend?

- A friend is a function that has accessed to a member of a class, but that function is not a member of that class.
- A friend function can be a member function of another class or any single function.
- The keyword `friend` is used to declare a friend in the function prototype

Example of friend function definition

```
class Watch{
      int hour;
      int minute;
      int second;
  public:
    Watch(int the_hour, int the_minute, int second);
    void input();
    void output();
    friend bool equal(Watch time1, Watch time2);
};
```

What is a Memberwise Assignment?

- Memberwise assignment is where the = operator may be used to assign one object's data to another object.
- Can be used to initialize one object with another object's data
- Example of copying a member to another member :

```
instance2 = instance1;
```

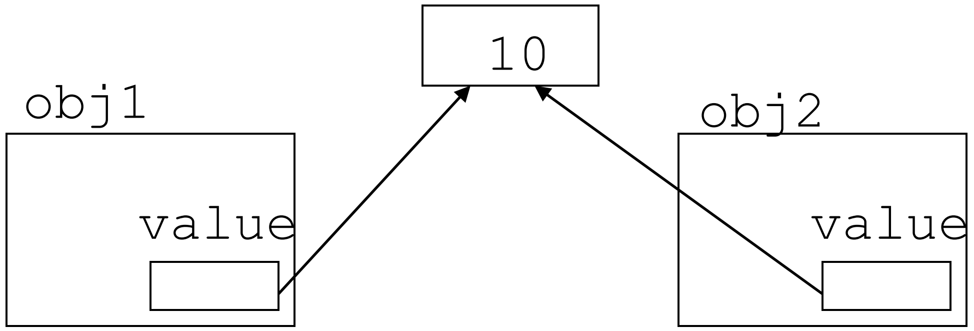
What is a Copy Constructors?

- Copy constructor is a special constructor to create new object and initialize it with a data from another object of same class.
- Default copy constructor copies field-to-field

Effect of a Copy Constructors

The result of the memberwise copy with objects containing dynamic memory:

```
AClass obj1(4);
AClass obj2 = obj1;
obj2.setVal(10);       //value is set to 10
cout << obj1.getVal(); //value is also 10
```



What is Operator Overloading?

- Operator overloading allows C++ programmer to redefine standard operators function when using class objects.
- To overload the + operator:

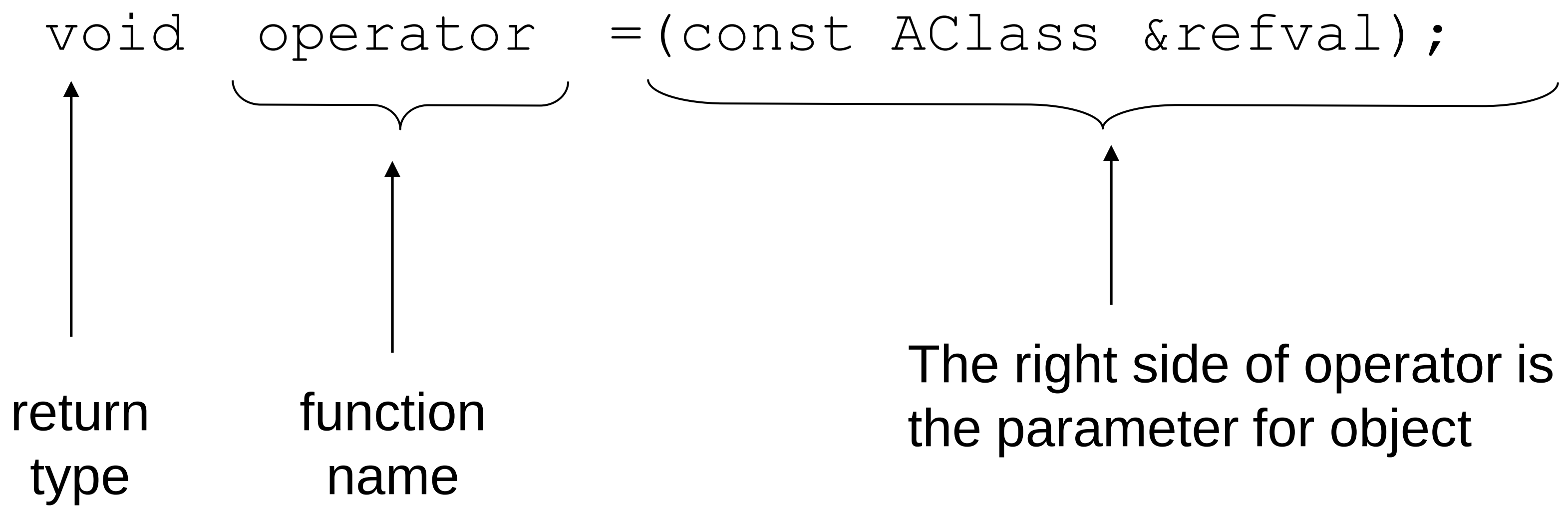
```
operator+
```

- To overload the = operator:

```
operator=
```

Operator Overloading

- Prototype:



```
void  operator  =(const AClass &refval);
```

- Operator is called via object on left side

Invoking an Overloaded Operator

- Call the operator as a member function:

```
obj1.operator=(obj2);
```

- Use operator in conventional manner:

```
obj1 = obj2;
```