Exercises

Constructors, Destructors, and Operators

Prof. Sai-Keung Wong

TA: xyz

One hour

This set of exercises does not have answers.

The students should work out the answers.

Basics

What is a constructor?

Define a class which has a constructor which has two parameters.

Basics

What is a destructor?

Define a class which has a constructor and a destructor. The destructor releases the memory space pointed by a member variable.

Basics

What is shallow copy?

Define a class and write a piece of code which invokes shallow copy.

Basics

What is deep copy?

Define a class and write a piece of code which invokes deep copy.

Basics

What is a copy constructor?

Define a class and write a piece of code which invokes the copy constructor.

Operators

Operators

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;     // greater than
        const A &operator = (const A &x);                // assignment
}
```

What is the meaning of const for the formal parameter A?

Operators

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;             // greater than
        const A &operator = (const A &x);               // assignment
}
```

What is the meaning of const at the end of the declaration of >?

bool operator > (const A &x) const

Operators

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;          // greater than
        const A &operator = (const A &x);            // assignment
}
```

What is the meaning of & for the declaration of the = operator?

```
const A &operator = (const A &x);
```

Operators

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;             // greater than
        const A &operator = (const A &x);               // assignment
}
```

Define the operators. Give one example to use each operator.

a and b are objects of A.

a > b iff a.v > b.v

Question

```
class A {
        protected:
        int v;                                          // value
        public:
        // addition. Return an object and its value is the sum of the values of the two operands.
        A operator + (const A &x) const;

        bool operator > (const A &x) const;    // greater than

        const A &operator = (const A &x);      // assignment
}
```

Define the operators. Give one example to use each operator.

a and b are objects of A.

a > b iff a.v > b.v

Question

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;      // greater than
        const A &operator = (const A &x);                  // assignment
}
```

For the > operator, can we define it as
bool operator = (const A &x)?

Explain your answer in details.

Question

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;     // greater than
        const A &operator = (const A &x);               // assignment
}
```

For the assignment operator, can we define it as
const A &operator = (const A &x) const?

Explain your answer in details.

Question

```
class A {
        protected:
        int v;
        public:

        bool operator > (const A &x) const;     // greater than
        const A &operator = (const A &x);               // assignment
}
```

For the assignment operator, can we define it as
const A operator = (const A &x)?

Explain your answer in details.

Question

```
class A {
	protected:
	int v;						// value
	public:

	// addition. Return the object
	// and its value is the sum of the values of the two operands.
	A &operator += (const A &x);
}
```

Define the operator +=.

Give one example to use it.

Question

```
class A {
        protected:
        int v;                                  // value
        public:

        // addition. Return the object
        // and its value is the sum of the values of the two operands.
        A operator -( ) const;
}
```

Define the unary operator -.

Return an object whose value is the negative of the operand’s value.

Give one example to use it.

Question

```
class A {
        protected:
        int v;                                  // value
        public:

        //prefix operator
        A &operator ++( );
}
```

Define the prefix operator ++.

Give one example to use it.

Question

```
class A {
        protected:
        int v;                                  // value
        public:

        //postfix operator
        A &operator ++( int dummy);
}
```

Define the postfix operator ++.

Give one example to use it.

Question

When should we use friend?
Define a class.
Then give one example to use friend.

Why do specify a function as friend in class?

- Private or protected members of a class cannot be accessed from outside of the class.
- Allow some trusted functions and classes to access a class's private and protected members.
- How do we use friend? Give two examples. Describe clearly the purpose of each example.

Answer

When should we use friend?
Define a class.
Then give one example to use friend.

```
class A {
protected:
 int v;
public:
  A(): v(0) { }
  friend ostream& operator<<(ostream& out, const A& x) {
    out << x.v;
    return out;
  }
};
```

Question

Define a class.

Convert an object of the class to a double.

Give an example to do the type casting for converting the object into a double.