

OOP with C++

Date 16/July

* Need of Object Oriented Programming?

C++ & C level language is basically Middle Level Language

C language supports structural approach.

C++ is an Objected Oriented Programming Language.

- C is less secure but C++ is more secure.

- C → Procedural

- C++ → Procedural (statements are structured in procedures)

- |→ Object Oriented

- |→ Generic Programming (templates)

- C follows top down approach

- C++ follows bottom up approach.

Difference?

OS [20 KB]

Processor [20 MB]

933 MHz → Processor speed

2GB → HDD

64MB → RAM

* Module: It is the collection of instructions

entity is programmatic

* First Object Oriented Programming language: SIMULA 67

(Hierarchy of Programming Language)

BASIC

B, C

Simula 67

C++

Java

Python

• DLL → dynamic link libraries

STL → storage template library.

C++ does not use any format specifier, e.g. %d, %f

Structure of C++ Prog.

Header file →

main()

of

function logic

}

① `#include <iostream>` → * Header file

Preprocessor Directive using namespace std; →
int main() → * put return value to
the main & to OS.

int a, b, sum;

cout << "Enter value a";

cin >> a;

cout << "Enter value b";

cin >> b;

sum = a + b;

cout << "Result";

cout << sum;

return 0;

}

→ console out

cout << "Enter a & b";

cin >> a >> b;

} cout << "Result" << sum;

printf → function

cout → object

cout : predefined object of iostream class.

cin : predefined object of istream class.

Spiral

- * \gg : Insertion operator or put to operator. (screen)
- * \ll : extraction operator or get from operator (keyboard)
- * Buffer : small amount of memory.
- writing too many operator in a single line is called Cascading of operators. (refers to consecutive occurrence of input & output in a single statement)

Functions in C++

- ① Call by Value
- ② Call by Reference(?)
- ③ Call by Address

* Types of variables:

- i) Normal Variable
- ii) Pointer Variable.

* In C++ a new variable called 'Reference Variable'.

* A reference variable is Alias for a normal variable.

② int $x = 10;$

int $\&p = x;$

① void swap(int , int);

void main()

swap(a,b);

void swap(int p, int q)

int temp;

temp = p

p = q

q = temp;

Call by
Value

③ void swap(int *, int *)

void main()

swap(&a,&b)

void swap(int *p, int *q)

Call by

Address

int temp;

temp = *p;

*p = *q;

*q = temp;

Spiral

Reference Variable: (External Pointer)

Date.....

Don'ts

int & x = 10;

int &p = x;

p++;

① int &p = 10; X

② int &p[5] = x; X

③ static int &p = x; X

→ swap (int &, int &)

void main ()

swap (a, b)

}

swap (int &p, int &q)

{

int temp;

temp = p;

p = q;

q = temp;

}

Notes: Swap is a predefined fun
is C++ so use swap!

Q: Tower of Hanoi in Recursion?
Exponential time complexity 2^n

Function Overloading in C++ → (Sup)

Creating function ~~with~~ having same task but having ~~different~~ different signatures. (Polymorphism)

void add (int, int)

int add (float, int)

void add (int, float, int)

Type of Parameters,

order,

no. of

parameters.

* Signature:

internally these parameter
bind by Mangling.

① Type of parameter

② Order of Order

③ no. of parameter

Date 23/July

(Q1) Given the coefficients of the quadratic poly " of type float.

Using C++ program, find whether the roots are real or complex. If the roots are real find them otherwise print the message "no. "real roots".

(Q2) An electricity board charges the following rates to its domestic customer for the consumption of energy.

✓ for 1st 100 unit = 60 paise/unit

✓ for next 200 unit = 80 paise/unit

✓ beyond 300 unit = 90 paise/unit

All users are charged a minimum of ₹ 50. If the total amount is more than ₹ 300 then an additional surcharge of 15% is added. ~~use AP in C++ to read the name of user & no. of unit consume & also display the charges with name.~~

Soln: ① If $(b^2 - 4ac) > 0$; root₁ = $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

root₂ = $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$

② If $(b^2 - 4ac) = 0$; root₁ = root₂ = $-b / 2a$

③ If $(b^2 - 4ac) < 0$; root₁ = $\frac{-b + i\sqrt{-(b^2 - 4ac)}}{2a}$

root₂ = $\frac{-b - i\sqrt{-(b^2 - 4ac)}}{2a}$

Default Arguments in C++

Date.....

e.g. `add (int x, int y, int z=0, int k)`

binding pairs.

`s = add (10, 20)`

* binding pairs are necessary in case of having arguments

e.g. `add (int k, int y=0, int z, int k=0)` X

binding should be in order only.

INLINE FUNCTION (Property): Compiler inserts the entire

(Q) How does DataBase works? body of the fun in the place where inline fun name is used in the program

A: consistency

ACID Property . e.g. Bank

I: isolation

D: durability

Property

(Q) Write a program in C++, to implement function overloading where the name of the function are.

Calculate ① Mr. of A

② Mr. of O = floor : 0.2 (mod 3) { }

③ Mr. of \square

(Top Left) - 1 = 1 - 1 = 0 floor

(A) WAP in C++ to input n no. & take odd no. & even no. from the inputed no. & print them sequentially.

Strings in C++

An array of character is called String.

C

C++

- | | |
|--|--|
| ① In C we cannot use arithmetic operations for string. | ① Arithmetic operators can be used for <u>string</u> .
In C++ <u>string</u> is designated as a class. |
|--|--|

e.g. #include <string>

or

#include <cstring>

```
int main() {
```

}

string str1 = "Hello";

string str2 = "World";

str2 = str1 + str2;

cout << str2;

}

e.g.

#include <cstring>

```
int main()
```

{

string str1 = "Hello";

string str2;

str2 = str1;

cout << str2;

}

Spiral

Scanned by CamScanner

C++ built-in

- (1) assign()
- (2) append()
- (3) insert()
- (4) comp()
- (5) find()
- (6) rfind()
- (7) replace()
- (8) erase()
- (9) substring()
- (10) length()
- (11) size()
- (12) reverse()
- (13) at()

C++ → string class

character array

e.g. ✓ str2 = str1;

or

str2.assign(str1)

e.g. str.erase(6, 2);

str1 = Hello Graphic

"Hello" = str1 ↳ "Hello aphe"

e.g. ✓ str1 = 'Hello'

str2 = 'world'

str2 = str1 + str2

or

str2.append(str1)

* it will start from
position 6 & it will
erase next to index.

e.g. ✓ str1.insert(3, "ABC");

str1 = 'Hello Graphic'

or ↳ "Hello ABC Graphic"

e.g. str1 = 'Hello Graphic Era University'

str1.find ("University");

String Function in C++

In C++ string is a 'special datatype'.

e.g. str = Hello. Graphic Era University

str.replace (6, 3, "xyz");

Output: Hello xyzGraphic Era University.

e.g. str.substr (9, 4)

Output: Graphic Era University.

e.g. str.at (6)

* Pass the ~~too~~ ~~char~~ index number and it will return the existing value.

Output: G

INLINE FUNCTIONS

Date..... 30/July

- # Define function is only a request to compiler.
- # A function that expands in a line.
- # Inline function is compiler dependent.

e.g.-

```
void print()
```

{

```
cout<<"Hello";
```

}

}

execution time 0.3ms

```
inline void print()
```

{

```
cout<<"Hello";
```

}

}

execution time 0.001ms

- * Including inline before void reduce the execution, it is compiler dependent whether he want to accept the inline function or not.
- It is a request not a command.

⇒ Don't for Inline functions:-

- recursive
- nested loops
- static variable
- switch-case
- while-condition
- multiple if-else.

- * Inline function when applied to function, it creates a copy of main in calling statement every time ~~#~~ the function calling the main.

* Anti matter

[* Up hand research]

Spiral

* for (char i=0; i<n; i++)
* for (int i=0; i<n; i++)

Date.....

10: WAP in C++ to find whether a given string is palindrome or not using string class?

20: WAP to implement function of Inline function of your own.

30: WAP in C++ to input a string and count the number of words in the given string, 1. count the number of pair vowels in the given string., 2. count the number of Palindrome words., 3. count the occurrence of each character., 4. Remove duplicate character from the string.

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

int main ()

{

string str, str1;

getline (cin, str)

str1 = str;

str. reverse (str);

if (str1 == str)

cout << "Palindrome" ;

else

cout << "Not Palindrome" ;

}

CLASS & OBJECT

Date 3/Aug

- ① **Class:** It is a blueprint, that defines behaviour of objects.
 ② **Object:** Real world entity.

* The building block of C++ that leads to OOP is a class.

It is a user defined datatype, which hold its own data datamember and member functions, which can be accessed and used by creating an instance of that class.

Specification of a Class: / Syntax

→ first letter

class <Class name>

↑
keyword

→ any meaningful name for your class.

class
Bottom up
Top down
struct

* All the datamembers in structure are by default public.

* All the datamembers in a class are by default private.

class <Class name>

↑

Data member 1 ;

Data member 2 ;

Data member 3 ;

public:

Member. f1();

Member. f2();

Member. f3();

};

e.g. #include <iostream>
using namespace std;

class geeks

Public:

string geekname; // member variable

void printname(); // member function

{ cout << "Geekname is" << geekname;

}

};

int main()

{

geeks obj; // declare an object of class

obj. geekname = "Abhi"; // access data member

obj1.printname(); // calling member function

return 0;

}

Spiral

Syntax of our object:

main()

{<class.name> <object.name>;} *Each class has atleast one main function.*
class name of class is mentioned before object name.
Object name can be anything but class name.
Object name must be unique in a class.
Object name must be unique in a program.

e.g. Class Test

{

 int a, b;

public:

 void set (int x, int y);

a = x;
b = y;

 void print();

{

 cout << "Value of a & b" << a << " " << b;

int main()

{

 Test ob;

 ob.set(100, 200);

 ob.print();

}

Accessing the member function

scope resolution operator
(membership label)

<return type> <class.name>::<function.name>

void Test :: set (int x, int y)

Q2. WAP to create a class student that contains the following parameters else the attributes

(i) name of student

(ii) enrollment no.

(iii) department

(iv) University

(v) Marks in 5 subjects

(vi) Address of student.

Create the necessary member functions getInfo()

calcMarks()

ShowInfo()

with the help of object, invoke the necessary details of the student, calculate the average & the % of the student, display all the details of the students.

Sol:

#include <iostream>

using namespace std;

class student

int i, marks[5], sum=0, av=0, enroll, num;

string name, department, university, address;

void getInfo();

int calcMarks();

void showInfo();

3;

Spiral

```
void student :: get_info()
```

```

cout << "Enter name : ";
cin >> name;
cout << "Enter department : ";
cin >> department;
cout << "Enter university : ";
cin >> university;
cout << "Enter address : ";
cin >> address;
cout << "Enter marks of 5 subject : ";
for (int i=0; i<n; i++)
    cin >> marks[i];

```

```

    cout << "Subject 1 mark : ";
    cout << "Subject 2 mark : ";
    cout << "Subject 3 mark : ";
    cout << "Subject 4 mark : ";
    cout << "Subject 5 mark : ";

```

```
int student :: calc_marks() {
    int sum = 0;
    for (int i=0; i<n; i++)
```

```

        sum = sum + marks[i];
    }
    float av = sum / n;
    return (av);
}
```

```
av = sum / n;
```

```
return (av);
```

```
void student :: show_info()
```

```

cout << "Name : " << name << endl;
cout << "Department : " << department << endl;
cout << "University : " << university << endl;
cout << "Address : " << address << endl;
cout << "Average marks : " << av << endl;
}
```

Spiral

Scanned by CamScanner

```

int main()
{
    inf cur's;
    student obj;
    obj.getinfo();
    cur = obj.cal-marks();
    obj.showinfo();
}

```

3. In above code; we have not taken any object
of class first standard. Value of obj will
automatically be zero because we are not
initializing.

so we can't print value of obj because it has not
been initialized.

So, we can't do this & this is known as
undeclared variable error.
It is also known as undeclared identifier error.
=> It means we have not declared any
variables & correctly written (Missing function).

Ex. 1)

- 1. We have to declare variable in main function.
- 2. applyinfo() is a function which takes
cur as argument & returns value
of marks();
- 3. Marks is a variable which is
declared in applyinfo() function.

Ex. 2)

- 1. We have to declare variable in main function.
- 2. applyinfo() is a function which takes
cur as argument & returns value
of marks();
- 3. Marks is a variable which is
declared in applyinfo() function.

O: WAP in C++, create a class employ having the parameters name of employee, department, employ id, basic salary, designation etc. Create the following functions:

- (i) getmp_info()
- (ii) cal_sal()
- (iii) show_info()

Create three member funt are responsible read the info of employ, calculate total salary of employ & displaying an overall info of employ.

$$\text{Total-Sal} = \text{basic_salary} + \text{HRA} + \text{TA} + \text{DA} + \text{Other Allowance}$$

$$\text{HRA} = 20\% \text{ of basic_sal}$$

$$\text{TA} = 10\% \text{ of basic_sal}$$

$$\text{DA} = 7.5\% \text{ of basic_sal}$$

Assuming Other Allowance $\neq 2000$.

Code: #include <iostream>

using namespace std;

int main();

class employ

{ public:

int empid, salary;

string employ;

string depart;

string design;

void getmp_info();

int cal_sal();

show_info();

```

void employ :: getemp_info()
{
    cout << "Enter name of employ : ";
    getline (cin, employ);
    cout << "Enter salary : ";
    cin >> salary;
    cout << "Enter department : ";
    getline (cin, depart);
    cout << "Enter employ Id : ";
    cin >> empid;
    cout << "Enter designation : ";
    getline (cin, design);
}

```

```

void employ :: cal_sal (int x)
{

```

```

    int total;

```

$$\text{Total} = x + (0.2 \times x) + (0.1 \times x) + (0.75 \times x) + 2000;$$

```

    return total;
}

```

```

void employ :: show_info()
{

```

```

    cout << "Employee name : ";

```

```

    cout << employ;

```

```

    cout << "Employee salary : ";

```

```

    cout << salary;

```

```

    cout << "Employ department : ";

```

```

    cout << depart;

```

```

    cout << "Employ designation : ";

```

```

    cout << design;

```

```

    cout << "Total Salary : ";

```

```

    cout << total;
}

```

Date: _____

```
info main()
{
    int total_salary;
    employ e;
    e.getemp_sal();
    total_salary = e.cal_sal();
    e.show_info();
}
```

STATIC VARIABLE

(8/Aug)

void show

```

{
    int f;
    static int k;
}
```

* ~~Static~~ value initialize with 0, (default variable)

Class/Object

static Data Variable

static method.

[Static means without any class.]

Spiral

```
class Bank
```

{

public:

int balance; → instance variable (depends on object)

static float roi; → class variable (scope within the class is same)

void setData (int b)

{

balance = b;

?

void setRoi (float r),

?

roi = r;

3

main ()

bank ob;

ob.setData (7.2f)

#include <stdio.h>

int fun();

int main()

< printf ("%d", fun());

printf ("%d", fun());

return 0;

int fun()

& static int count = 0;

count++;

return count;

(Bank::) setRoi (8.2f)

3

→ static fun is capable of accessing static data variable only.

* Instance member fun can access static data variable.

* Static variable can be only be declared inside the class.

* Static variables must be defined outside the class.

* memory is allocated to static variable.

=> <return-type> <class-name> :: <name of variables>

Constructors in C++

Q* what is a constructor?

- => • It is used to solve the problem of initialisation
- It is a special member function of a class that is used to solve the problem of initialisation.
- It is completely dependent on object creation.

Properties:

- i) Name of the constructor should be same as that of a class.
- ii) It does not have return type.
- iii) It is not static.

How to Create a Constructor?

Class Test

{

 Public:

 Test () // constructor created (implicitly)

 {

 cout << "Constructor Invoked" ;

}

};

Spiral

main()

{

Test ob;

ob. Test();

}

created

* Constructor (implicitly) by the compiler.

* we can also make invoke if we define a constructor

⇒ class Test

{

public:

Test()

{

cout << "Constructor";

}

};

main()

{

Test ob1, ob2, ob3, ob4; or _____

}

* constructor does not require dot(.) operator

* if we create any constructor by user the parameter need to be declared.

⇒ ↗

Three Types of Constructors:-

① Default Constructor.

② Parameterized Constructor,

③ Copy Constructor.

Spiral

Class Test

{

put a, b;

Public:

Test (put x, put y)

{

a = x;

b = y;

cout << a << b;

}

Test (int k)

{

a = k;

cout << a;

}

};

main()

{

Test ob(100, 200) ob(400), ob;

}

* All the methods in C++ will always be declared as
 Public.

~~Q1. WAP in C++~~

(Q1: WAP to create a class employ containing parameters.

- Joining
- Employ ID
- Employ name
- Department
- salary.

Create a function that is constructor for reading the data-variables of a class. Create another fun

to print the contents/info. of employ with the help of object.

(Q2: Create a class student containing the parameters

- name
- department
- Enrollment no.
- University
- Age

Out of these parameters qualify age & enrollment number as static. Create a member that is used to access these members also create a method that is responsible for accessing the non-static members of class.

Soln: class employ

```

int empid, salary;
char name[20], dept[20];
float sal;
```

Public :

employ() // constructor

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

class employ {
    protected:
        int emp_id;
        string name;
        string department;
        float salary;
    public:
        void getdata (cin, name);
        void putdata (cout);
        void display ();
};

void employ :: getdata (cin, name) {
    cout << "Enter Employee id : ";
    cin >> emp_id;
    cout << "Enter Name : ";
    getline (cin, name);
    cout << "Enter Department : ";
    cout << "Enter salary : ";
    cin >> salary;
}

void employ :: putdata (cout) {
    cout << "Enter Employee id : ";
    cout << emp_id;
    cout << endl;
    cout << "Name of employ : ";
    cout << name;
    cout << endl;
    cout << "Department : ";
    cout << department;
    cout << endl;
    cout << "Salary : ";
    cout << salary;
}

void employ :: display () {
    cout << "Employee Id : ";
    cout << emp_id;
    cout << endl;
    cout << "Name of employ : ";
    cout << name;
    cout << endl;
    cout << "Department : ";
    cout << department;
    cout << endl;
    cout << "Salary : ";
    cout << salary;
}

int main () {
    employ obj;
    obj.getdata (cin, name);
    obj.putdata (cout);
    obj.display ();
}
```

Date 22/April/1

Qs: A book shop is responsible for maintaining the books that are being sold at the shop. (It includes the details such as author's name, book title, book price, publisher's name and stock position). Whenever a customer wants a book the sales person inputs the title and the author's name & the system searches the book and displays whether it is available or not. If it is not available, an appropriate msg is displayed. If available then the system displays the book details & requests for the no. of copies required. If the requested copies are available then the total cost of the requested copies is displayed otherwise the system will print the message 'Requested Copies not available'. Design a system using the concept of class and obj & take necessary member fun/constructors for this program.

class book

{

```
string author_name, book_title, publisher_name;  
int book_price, stock_position, copies;
```

Public :

```
void get_info();  
void show_info();  
int find();
```

} b;

143h c c

Date.....

void book :: get_info()

cout << "Enter book title:";
getline (cin, book-title);
cout << "Enter author:";
getline (cin, author-name);

}

int book :: find()

if (book-title == book || author-name == name) cout << "We are having" << copy << "copies";

{

cout << "We are having" << copy << "copies";

}

else

{

cout << "We do not have any stock";

}

Q2: WAP to create class student & print N student details.
 Student details are: name, roll no, section, marks of 5 subjects,
 percentage, grade.

If $\% > 60$ - A

$60 \leq \% < 50$ - B

Class Test

{

int a;

public:

Test (int value) // Parameterised Constructor.

{

a = value;

}

Test ()

{

void print()

{

cout << "Value of a " << a;

.

};

int main()

{

Test ob, ob1(100), ob2(ob1); // copy constructor

- ob1.print();

ob2.print();

}

In a program, if we don't use any constructor (object creation)
 then compiler will create two constructor (default, copy).

- If user creates/defines explicitly then default constructor of the compiler will not be executed but copy constructor is still working until unless the user creates/ defines its own copy constructor explicitly.

user defined constructor

Test (Test &t) // copy constructor

a = t.a; // ob2 gets the value of copy constructor

e.g. Test ob1(100), ob2(ob1)

or

Test ob1(100), ob2

ob2 = ob1;

or

Test ob1(100)

Test ob2 = ob1;

Q: Create a class time that has separate data members for hrs., min, & sec. of type int. Create a constructor that initializes those data members to zero & another constructor should initialize it to some fixed values.

The output invoked the member function should display the time in this format 11:59:59. A member function should be created having name add, that is used to add two objects of type time passed as arguments. Create a function main() that contains these two initialized values together, leaving the result in third time variable. Finally it should display the value of third time variable.

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

```
class time
```

```
    int hr, min, sec;
```

```
public:
```

```
time () {
```

```
    hr = 0;
```

```
    min = 0;
```

```
    sec = 0;
```

```
}
```

```
time (int a, int b, int c)
```

```
{
```

```
    int x, y, z;
```

```
    x = a / 60; hr = a;
```

```
    y = b % 60; min = b;
```

```
    z = c % 60; sec = c;
```

```

if (a >= 60)
    x = a / 60;
else if (b >= 60)
    y = b / 60;
    b = b + y;
    }
}

```

class Atime

```

private:
    int hr, min, sec;
public:
    time()
    {
        hr = 0;
        min = 0;
        sec = 0;
    }
}

```

`time (int hr, int min, int sec)`

```

int a = 0;
if (sec >= 60 && sec < 120)

```

```

    a = 1;
    sec = sec - 60;
    min = min + a;
    a = 0;
}
else if (sec >= 120)
{
    a = sec / 60;
    sec = sec - a * 60;
    min = min + a;
    a = 0;
}

```

```

if (min >= 60 && min < 120)

```

```

    a = 1;
    min = min - 60;
    hr = hr + a;
}
else if (min >= 120)

```

```

    a = min / 60;
    min = min - a * 60;
}
}

```

`if (a >= 60)`

```

x = a % 60;
hr = hr + x;
}
}

```

`if (b >= 60)`

```

y = b % 60;
min = min + y;
}
}

```

`if (c >= 60)`

```

z = c % 60;
sec = sec + z;
}
}

```

`hr = hr + a;``if (hr >= 12 && hr < 24)``a = 1;``hr = hr - 12;``else if (hr >= 24)``a = hr / 12;``hr = hr - a * 12;``void add (time t, time q)``sec = t.sec + q.sec;``min = t.min + q.min;``hr = t.hr + q.hr;``time (hr, min, sec);`

but we will,

q
 dme. ob1 (110, 50, 150);
 dme. ob2 (50, 80, 100);
 dme. ob3;
 ob3. add (ob1, ob2);
 ?

~~Ques~~

FRIEND FUNCTION IN C++

Q: What is friend function.

Q: Why it is used?

Ans: It is a (non-member function) that can access private/protected data member of a class directly.

- It is outside the scope of the for class or the scope of this for is not known to the class.
- Definition will be out of the class.
- It does not have any caller object.
- It must be declared within the class.

e.g. class test

int a, b;

public:

(Test f);
 friend void f(); // must be declared

void get (int x, int y)

a = x;

b = y;

?;

Spiral

int main()

1

```
Test obj;
obj.get(40, 80)
add(obj);
```

3

void add (test &t)

{

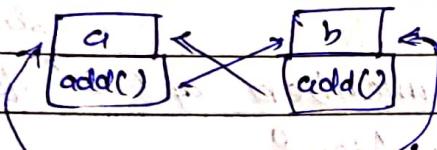
cout << "Sum is : << t.a + t.b; (addition of two numbers)

3

→ Direct access to a fb.

3' d' r' - member fun bts

*



friend add() (to check if it friend it can access the variables)

* hence friend fun for a non-member so we can also write it in public also.

class B; // Forward declaration

e.g.

class A

4

int a;

public:

void get (int x)

5

a = x;

6

friend void add(A&, B&);

7;

class B

8

int b;

public:

void get (int y)

9

b = y;

10

friend void add(A&, B&);

11;

int main ()

{

A ob1;
B ob2;
ob1.get(10);
ob2.get(20);
add(ob1, ob2);

}

void add(A ob1, B ob2)

{

cout << "Sum is : " << ob1.a + ob2.b;

Q1: WAP in C++ to swap ~~the~~ members of '2' class using the concept of friend function?

Q2: WAP in C++ that contains two diff. class containing different members, find the greatest of two numbers using friend function.

class B;

class A;

int a;

public:

void get(int x)

class B;

(a) int b;

public:

B::get(int y)

class B;

b;

friend void y(A&, B&)

3;

swap

friend void (A&, B&)

3;

swap

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

```
class B;
```

```
class A
```

```
{
```

```
    int a;
```

```
public:
```

```
    void get(int n)
```

```
{
```

```
    a = n;
```

```
}
```

```
friend void add(A&, B&);
```

```
}
```

```
class B
```

```
{
```

```
    int b;
```

```
public:
```

```
    void get (int j)
```

```
{
```

```
    b = j;
```

```
}
```

```
friend void add(A&, B&);
```

```
}
```

```
int main()
```

```
{
```

```
    A ob1;
```

```
    B ob2;
```

```
    ob1.get(10);
```

```
    ob2.get(20);
```

```
    add(ob1, ob2);
```

```
}
```

```
void add(A& ob1, B& ob2)
```

```
{
```

```
    cout << "Sum is : << ob1.a + ob2.b;"
```

Spiral

Scanned by CamScanner

#include <iostream>
using namespace std;

class B;

class A {

}

int a;

public:

void get(int x)

{

a = x;

}

friend void swap(A&, B&);

}

class B

{

int b;

public:

void get(int y)

{

b = y;

}

friend void swap(A&, B&);

}

int main()

{

A ob1;

B ob2;

ob1.get(10);

ob2.get(20);

swapi(obj1, obj2);

void swapi(A&obj1, B&obj2)

{

int temp;

cout << "Initial value : " << obj1.a << " " << obj2.b;

temp = obj1.a;

obj1.a = obj2.b;

obj2.b = temp;

cout << "\n Swap is : " << obj1.a << " " << obj2.b;

Destructor

Date 11/sept.

Destructor is used to deallocate the memory related to object.

- ⇒ no argument is passed inside destructor ~`~test()`.
- ⇒ it cannot be overloaded.
- ⇒ it cannot be static.
- ⇒ it cannot be inherited.

Operator Overloading in C++

compile time overloading.

Test

```
ob1, ob2, ob3;  
ob1.get(10, 20);  
ob2.get(30, 40);  
add = ob1.add(ob2);  
ob3.display();
```

Test

`add (test t)`

{
 test temp;

`temp.a = a + t.a`

`temp.b = b + t.b;`
`return (temp);`

2

Spiral

Test

```

obj, obj2, obj3;
obj1.get(10, 20);
obj2.get(20, 30)
obj3 = obj1 + obj2
obj3.display();

```

Test (operator +) (dest +)

```

temp temp;
temp.a = a + t.a;
temp.b = b + t.b;
return (temp)

```

Only those operator can be overloaded in C++ which are embedded in C.

ANSWER PAGE 4

* It includes pointer to object and not to variable
that means if object is made static then it
cannot be modified by any other function

Operator Overloading of Unary operator.

Unary

- negation
- Pre (Increment/Decrement)
- Post (Increment/Decrement)

e.g.: $x = ++y;$ $x = ?$ $y = ?$ $\boxed{x = 6}$ $x = y++;$ $x = 6$ $y = ?$

\Rightarrow (int) is the only variable which differentiate b/w pre/post operation.

[post increment (int)]

Q2: Implement operator overloading of binary operator +, -, *, /, pre increment & pre decrement, post increment, post decrement with or without friend function?

THIS POINTER

It is a local object point in every instance member for of a capable class that is capable of access the caller object.

Test th, *P.;

(t^*) get ()
↓
caller
object.

Spiral

It shows difference b/w instance variable and local variable.
Local always gets the priority.

[Instance]

```
int a,b;  
public:
```

```
void get (int a, b)
```

```
{  
    this.a = a;  
    this.b = b;  
}
```