Programming in C++ (Part-2)

ABOUT ME : NAVNEET GUPTA

- 8 years teaching experience.
- AIR 92 in GATE 2008
- Qualified UGC-NET 2012, Raj.-SET 2012, CSIR-Recruitment-Exam in 2011
- Achieved 3rd Rank in NPTEL-DBMS Course
- Achieved Silver Medal in CSIR on ERP Project in 2013
- Area of Expertise : DBMS, Programming, Algorithms, Discrete Maths, Computer Networks, Operating system

gradeup
Prep Smart. Score Better.

C++ Constructor

int a = 20; // Variable definition

(*) In c++, constructor is a special method which is invoked automatically (called) at the time of object creation.

(*) It is used to initialize the data members of new object generally. The constructors have the same name as of class name in c++.

Cl. — X

gradeup
Prep Smart. Score Better.

Types of Constructor

Two types of Constructors in c++

→ (i) Default Constructor

→ (ii) Parameterized constructor.

gradeup
Prep Smart. Score Better.

1) C++ Default Constructor

A Constructor which has no arguments is known as default constructor.

eg:

```
class emp
{
  public :
    emp ()
    {
      // cout << " default constructor " ;
    }
};
```

```
main ()
{
  emp e1;
  emp e2;
}
```

e.g.:

default Constructor

default Cons-tructor

gradeup
Prep Smart. Score Better.

C++ Parameterized Constructor

A constructor which has parameters is called parameterized constructor. It is used to provide different values to distinct objects.

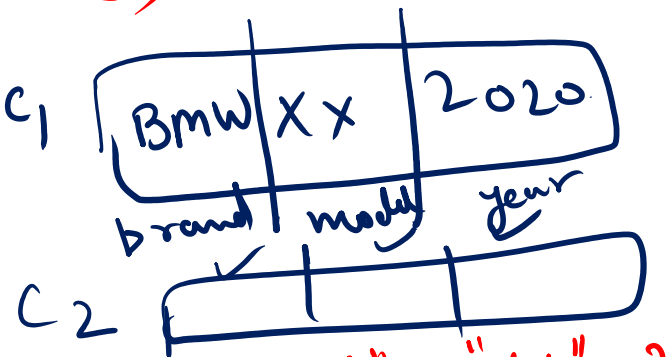
gradeup
Prep Smart. Score Better.

Example:

```
class Car
{
 public :
    string brand;
    string model;
    int year;

 Car (string b, string m, int y)
 { brand = b;
   model = m;
   year = y;
 }
};
main ()
 {
   Car c1("BMW", "XX", 2020);
   Car c2("Ford", "XY", 2019);

   cout << "Best & favourite
    car is " << c1.brand;
 }
```

data member

Param. Constructor



o/p

Best & favourite car is BMW

gradeup
Prep Smart. Score Better.

Just like functions, constructors can also be defined outside the class.

```
class Car
{public:
    string brand;
    string model;
    int year;
    Car(string b, string m, int n);
};

Car::Car(string x, string y, int z)
{
    brand = x;
    model = y;
    year = z;
}

main()
{
    Car c1("BMW", ....);
}
```

gradeup
Prep Smart. Score Better.

C++ Access Specifiers

*) Access specifiers defines how the members (attribute & method) of a class can be accessed.

*) In C++, we have 3 types of Access specifier:

(i) Public
(ii) Private
(iii) Protected

gradeup
Prep Smart. Score Better.

Example

emp e1, e2

1) Public: Members of the class are accessible from outside the class.

2) Private: Member of the class cannot be accessed outside of the class.

3) Protected: Members can be accessed outside the class but they can be accessed in the inherited class.

gradeup
Prep Smart. Score Better.

Example

```
class myclass
{
public:
    int x;
private:
    int y;
};
```

```
main()
{
    myclass obj1;
    obj1.x = 20;
    obj1.y = 50;
    cout << obj1.x << obj1.y;
}
```

Not allowed

gradeup
Prep Smart. Score Better.

Note:1 It is possible to access private members of a class using a public method inside the same class.

Note:2 By default, all members of a class are private if you don't specify an access specifier.

gradeup
Prep Smart. Score Better.

C++ Destructor

*) A destructor works opposite of constructor. It destructs object of classes. It can be defined only once in a class. Like constructor, it is called automatically.

(*) A destructor is defined like constructor. It must have same name as that of class but is predefined with tilde (~)

(*) C++ destructor cannot have parameter & also no return value & return type.

gradeup
Prep Smart. Score Better.

Example:

```
class str
{
  private :
    char *s;
    int size;
  public :
    str (char *);
    ~str();
};
```

```
str :: str (char *c)
{
  size = strlen(c);
  s = new char [size+1];
  strcpy (s, c);
}
str :: ~str()
{
  delete [] s;
}
```

```
main()
{
  str s1("Hello");
}
```

gradeup
Prep Smart. Score Better.

this Pointer

"this" is a keyword that refers to the current instance of the class.

(OR)

It can be used to pass the current object as a parameter to another method.

gradeup
Prep Smart. Score Better.

Example of this Pointer

```
class emp
{
    int salary;
    string name;
public:
    emp(int salary, string name)
    {
        this.salary = salary;
        this.name = name;
    }
```

data members

sal = sal

25000

Reena

```
void display()
{
    cout << salary << name;
}
main()
{
    emp e1(25000, "Reena");
    emp e2(100, "Solan");
    e1.display();
}
```

salary	name
25000	Reena

gradeup
Prep Smart. Score Better.

Friend function

*) If a function is defined as a friend function in c++ then the protected & private data of a class can be accessed using the function.

*) for accessing the data, the declaration of a friend function should be done inside the body of a class starting with the keyword "friend".

gradeup
Prep Smart. Score Better.

Declaration of friend function in C++

```
class classname
{
    friend datatype function-name (arguments);
};
```

gradeup
Prep Smart. Score Better.

Characteristics of a Friend function:

- The function is not in the scope of the class to which it has been declared as a friend.
- It cannot be called using the object as it is not in the scope of that class.
- It can be invoked like a normal function without using the object.
- It cannot access the member names directly and has to use an object name and dot membership operator with the member name.
- It can be declared either in the private or the public part.

gradeup
Prep Smart. Score Better.

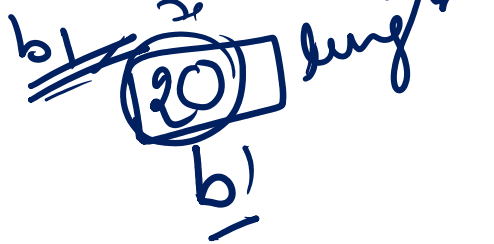
Example:

```
class box
{ int length;
  public:
    Box ()
    { length = 0; }
    friend int prtLgt (Box);
};

int prtLgt (Box b1)
{ b1.length = 20; }
  return (b1.length);
}

main ()
{
  Box b;
  cout << "Length of box = " << prtLgt (b);
}
```

default constructor

o/p.

Length of Box = 20



gradeup
Prep Smart. Score Better.

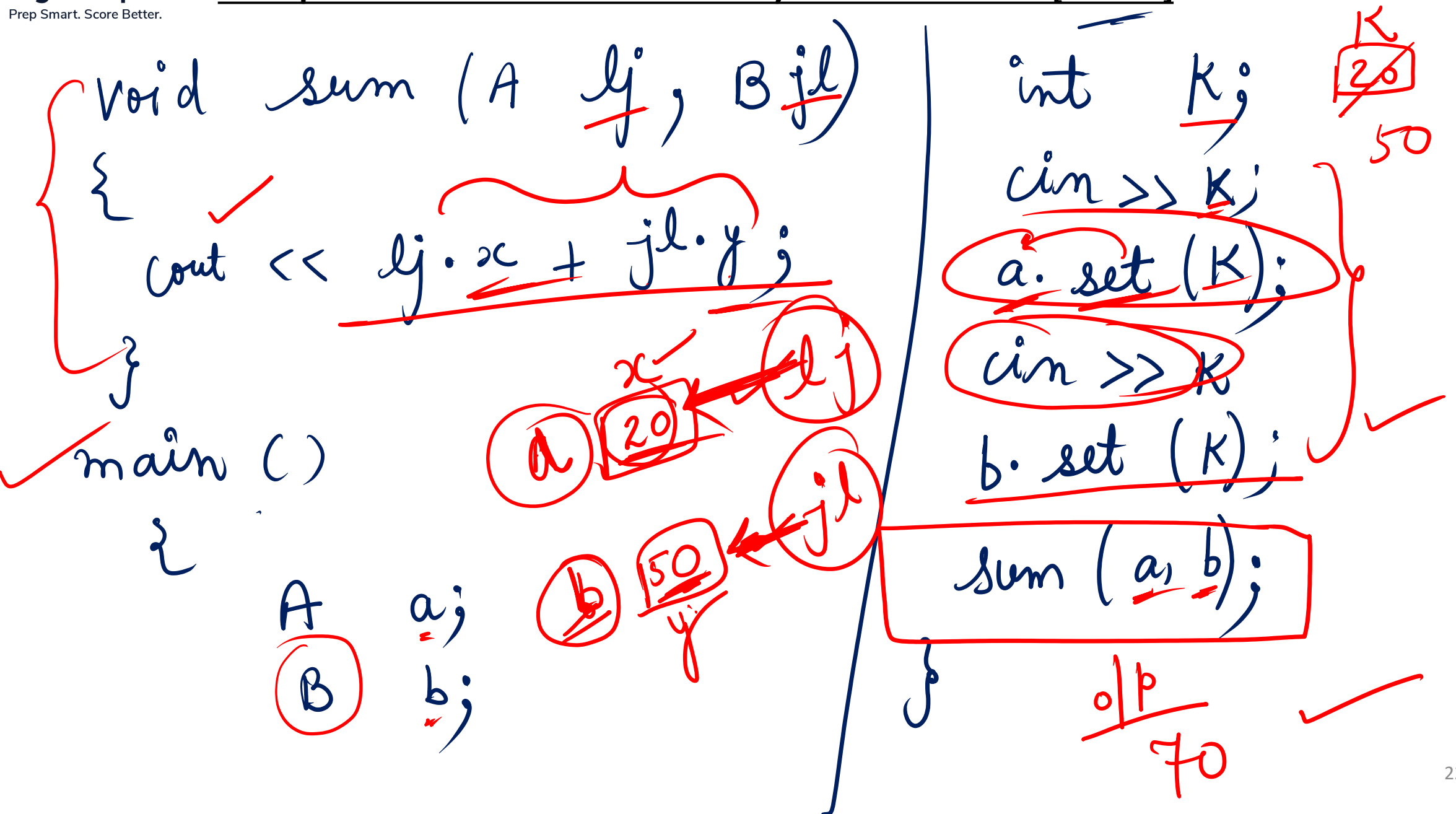
Example when the function is friendly to two classes.

```
class B;
class A
{
    int x;
public:
    void set (int p)
    {
        x = p;
    }
    friend void sum (A, B);
};
```

```
class B
{
    int y;
public:
    void set (int l)
    {
        y = l;
    }
    friend void sum (A, B);
};
```

gradeup
Prep Smart. Score Better.

Example when the function is friendly to two classes----[Contd.]



gradeup
Prep Smart. Score Better.

(UGCNET-DEC2016-II-11) Which of the following cannot be passed to a function in C++?

(1) Constant
(2) Structure
(3) Array
(4) Header file

gradeup
Prep Smart. Score Better.

3

(UGCNET-Dec2012-II-39) Match the following with respect to C++ data types :

a. User defined type	1. Qualifier
b. Built in type	2. Union
c. Derived type	3. Void
d. Long double	4. Pointer

Code :

a b c d

(A) 2 3 4 1

(B) 3 1 4 2

(C) 4 1 2 3

(D) 3 4 1 2

gradeup
Prep Smart. Score Better.

(UGC NET Paper-2 December 2007 No 15) Which of the following is true of constructor function in C++ ?

(A) A class must have at least one constructor.
(B) A constructor is a unique function which cannot be overloaded.
(C) A constructor function must be invoked with the object name.
(D) A constructor function is automatically invoked when an object is created.

gradeup
Prep Smart. Score Better.

5) (UGCNET-DEC2016-II-13) Which of the following storage classes have global visibility in C/C++?

(1) Auto ✗
(2) Extern ✓ : across multiple files.
(3) Static ✗
(4) Register ✗

gradeup
Prep Smart. Score Better.