OOPS Concepts And C++ Programming

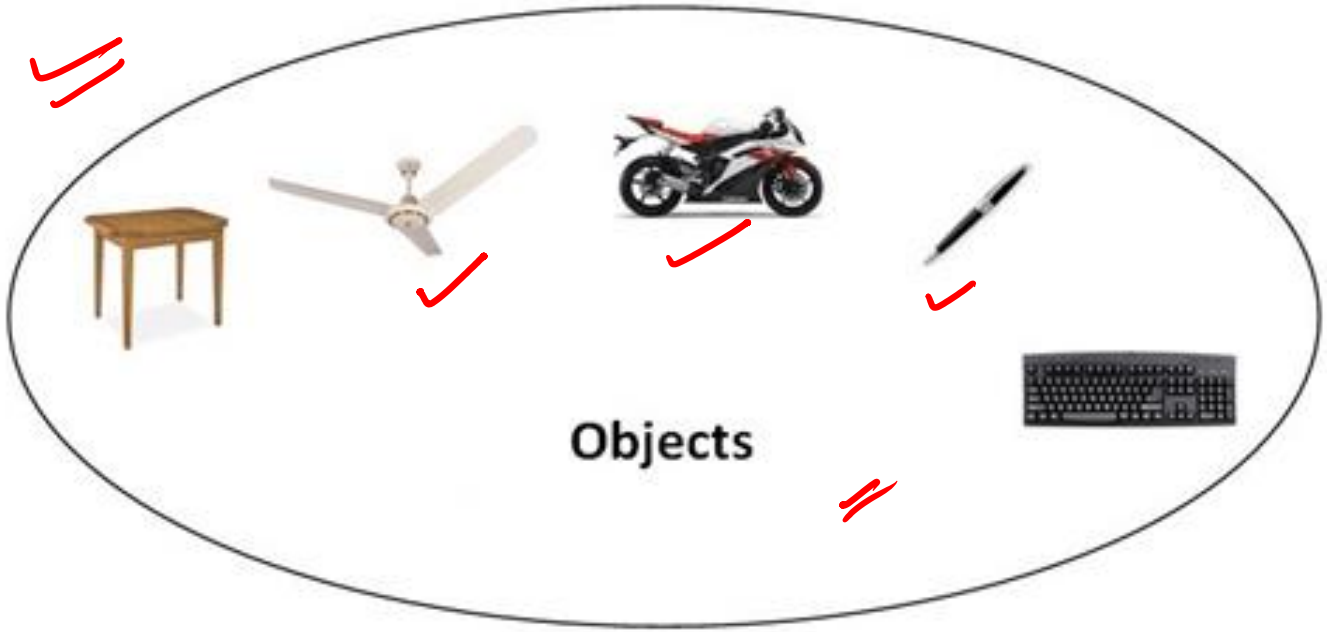
ABOUT ME : NAVNEET GUPTA

- 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.

OOPs (Object Oriented Programming System)

Object means a real word entity such as pen, chair, table etc. **Object-Oriented Programming** is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:



gradeup
Prep Smart. Score Better.

What are the four basics of object-oriented programming?

(i) Encapsulation

(ii) Abstraction

(iii) Inheritance

(iv) Polymorphism

gradeup
Prep Smart. Score Better.

Encapsulation

combining data & function together in one unit is known as encapsulation.

(*) Encapsulation in c++ is a mechanism for wrapping the data and code together as a single unit. In the encapsulation the variables of a class will be hidden from other classes & can be accessed only through the method of their class. Therefore it is also known as data hiding.

gradeup
Prep Smart. Score Better.

(11) Abstraction

eg: bike ✓ user. Ref. ✓

Abstraction is a process of hiding implementation details from the user, only the functionality will be provided to the user.

In other words, the user will have the information on what the object does instead of how it does it.

gradeup
Prep Smart. Score Better.

(iii) Inheritance !

Inheritance can be defined as process where one class (~~parent/subclass~~) can ~~override~~ inherits the functionality (method & fields) of another class.

(*) With the use of inheritance, the information is made managable in a hierarchical order.

gradeup
Prep Smart. Score Better.

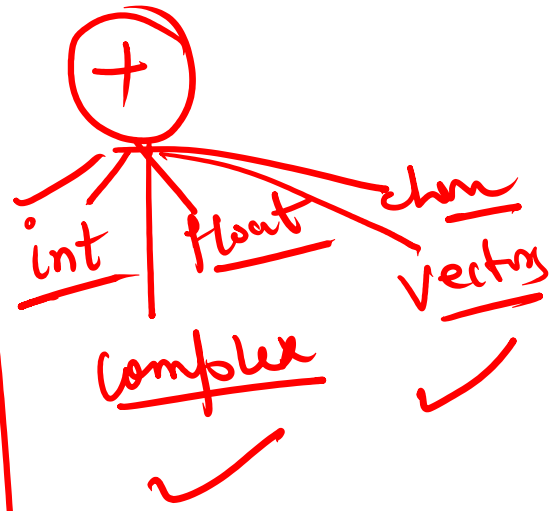
(iv) Polymorphism :

poly + morph

poly → more than one

morph → to change / convert

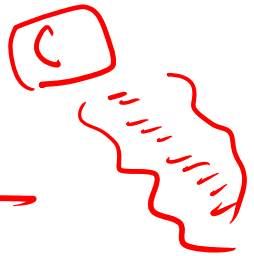
Human being →
class → student
family → child
friends → friend
→ brother



(*) Polymorphism is the ability of an object to perform different actions (or exhibit different behaviours) based on the context.

gradeup
Prep Smart. Score Better.

Advantage of OOPs over Procedure-oriented programming language



1. OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.

2. OOPs provide data hiding whereas in Procedure-oriented programming language a global data can be accessed from anywhere.

3. OOPs provide ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.

gradeup
Prep Smart. Score Better.

Object

(*) Any entity that has state & behaviour is known as object. eg chair, pen ...
It can be physical or logical.

(*) Object is a runtime entity. It is created at the run-time.

(*) Object is an instance of a class. (copy)

All the members (data members & member functions) of the class can be accessed through object.

gradeup
Prep Smart. Score Better.

Class

(*) Collection of objects is called class. It is a logical entity.

physical object → car, employee

logical " → a job

gradeup
Prep Smart. Score Better.

What is C++ programming?

Bjarne Stroustrup in 1979-1980

(*) c++ is a general-purpose, case-sensitive free-form programming language that supports object-oriented, procedural & generic programming

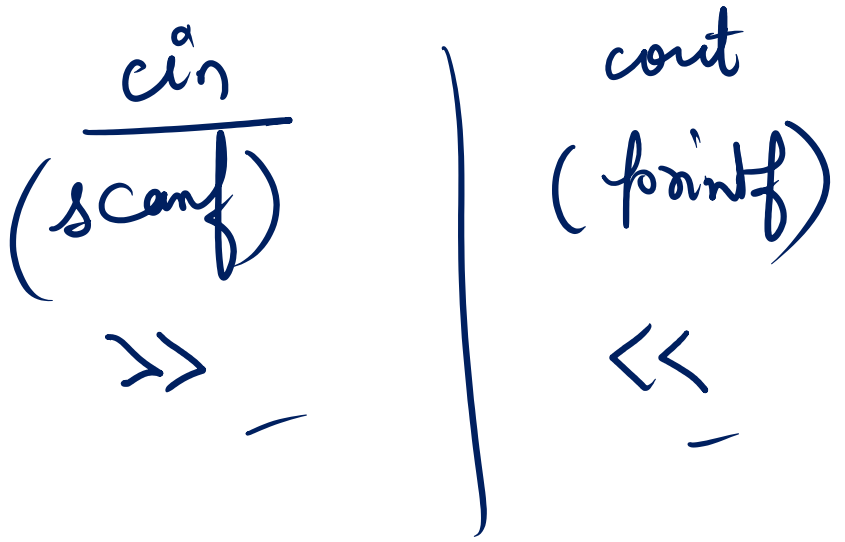
≈ while

(*) c++ is an object-oriented prog^n. It is an extension to c prog^n.

(*) c++ is a middle-level language as it encapsulates both high & low-level language features.

gradeup
Prep Smart. Score Better.

C++ Basic Input/Output



gradeup
Prep Smart. Score Better.

C++ Basic Input/Output

gradeup
Prep Smart. Score Better.

Standard output stream (cout)

(*) The cout is used in conjunction with stream insertion operator (<<) to display the output on a console.

gradeup
Prep Smart. Score Better.

Standard input stream (cin)

The cin is used in conjunction with stream extraction operator (>>) to read the input from console.

gradeup
Prep Smart. Score Better.

Example:-

```
#include <iostream>

int main()
{
    int age;
    cout << "Enter the age";
    cin >> age;
    cout << "The Age after 20 years is:" <<
        age+20;
}
```

Output

```
Enter the age
30
The Age after 20 years is: 50
```

gradeup
Prep Smart. Score Better.

Standard end line (endl)

(*) It is used to insert a new line characters & flushes the stream.

eg:
```
cout << "Gradeup" << endl;
cout << "Best one" << endl;
```

```
cout << "Hello"
cout << "Bye"
```

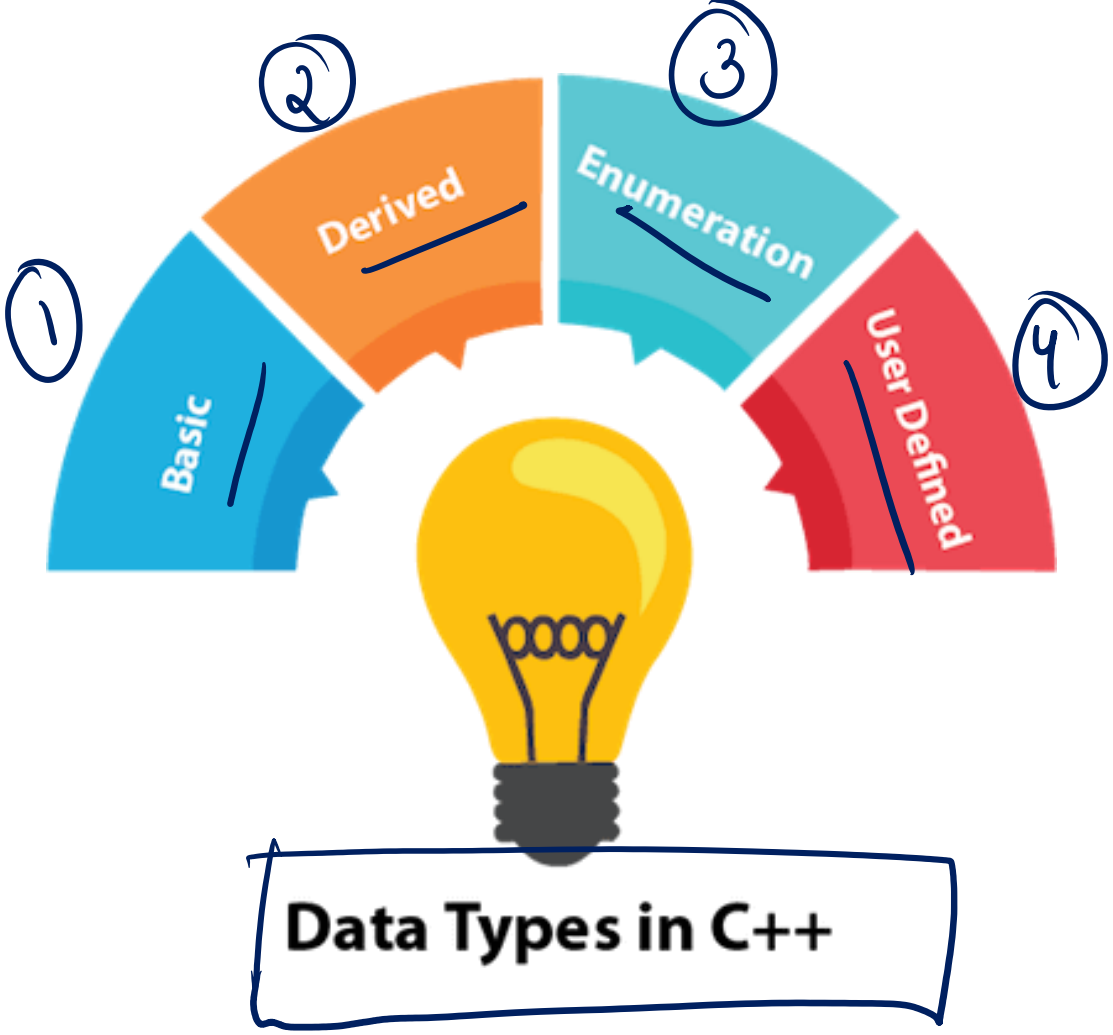
HelloBye

gradeup
Prep Smart. Score Better.

C++ Data Types

escape sequence in C language.

\n	\t	\b
\a	\\	\'



gradeup
Prep Smart. Score Better.

Stream
Flow of Characters.

keywords <= 60

Data Types	Memory Size	Range
char	1 byte	-128 to 127
signed char	1 byte	-128 to 127
unsigned char	1 byte	0 to 127
short	2 byte	-32,768 to 32,767
signed short	2 byte	-32,768 to 32,767
unsigned short	2 byte	0 to 32,767
int	2 byte	-32,768 to 32,767
signed int	2 byte	-32,768 to 32,767
unsigned int	2 byte	0 to 32,767
short int	2 byte	-32,768 to 32,767
signed short int	2 byte	-32,768 to 32,767
unsigned short int	2 byte	0 to 32,767
long int	4 byte	
signed long int	4 byte	-ve +ve
unsigned long int	4 byte	
float	4 byte	
double	8 byte	
long double	10 byte	

gradeup
Prep Smart. Score Better.

Example:

```
int myNum = 5;              // Integer (whole number)
float myFloatNum = 5.99;    // Floating point number
double myDoubleNum = 9.98;  // Floating point number
char myLetter = 'D';        // Character
bool myBoolean = true;      // Boolean
string myText = "Hello";    // String
```

gradeup
Prep Smart. Score Better.

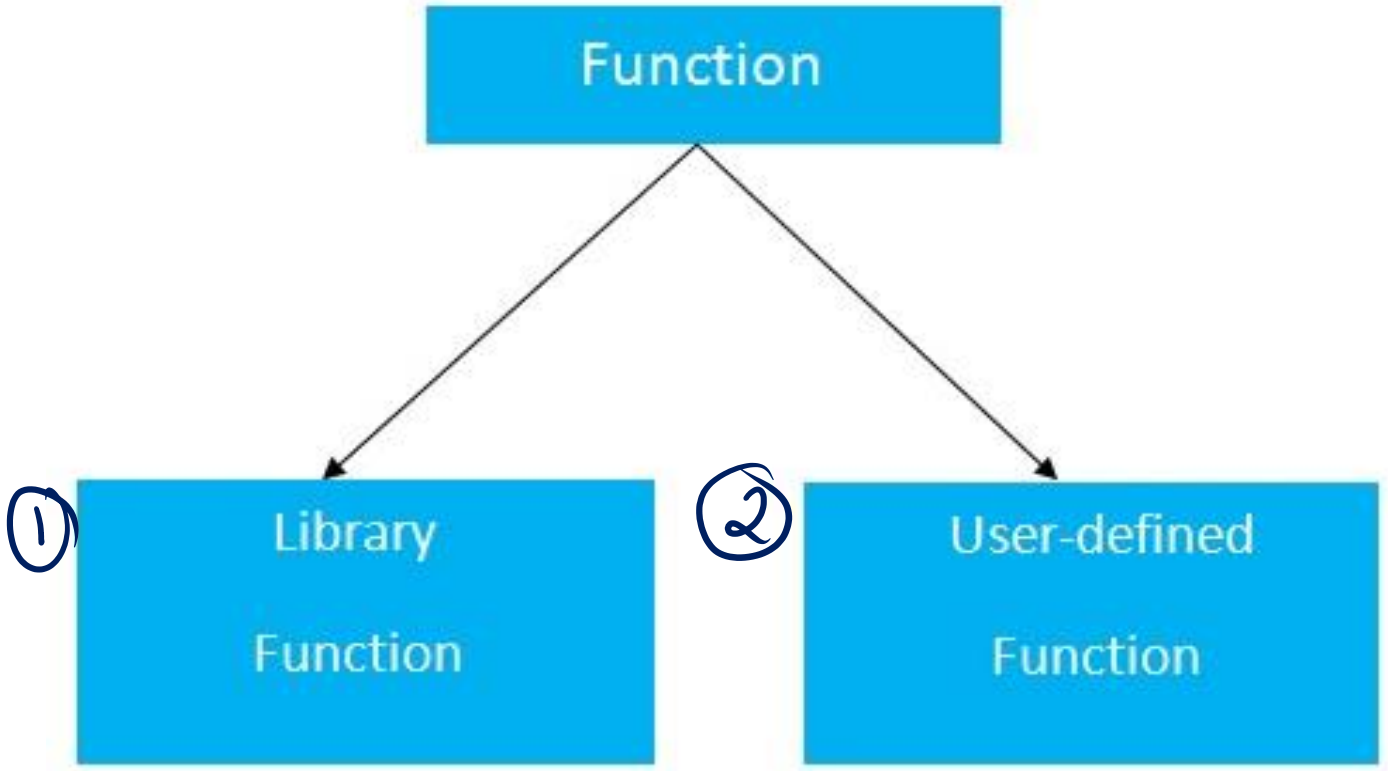
C++ Functions

1) predefined
2) user-defined

gradeup
Prep Smart. Score Better.

C++ Functions

Types of Functions



gradeup
Prep Smart. Score Better.

Example of Function:

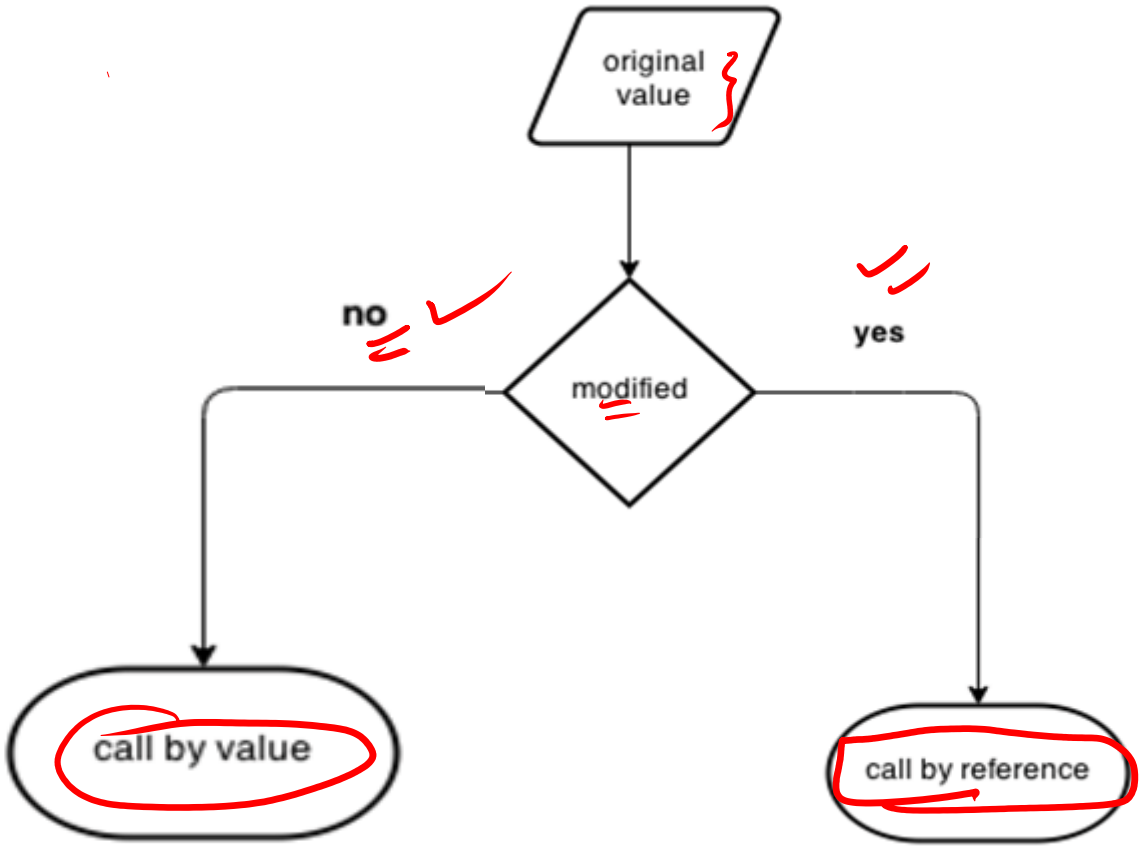
```
void func()
{
    static int i = 0;
    int J = 0;
    i++;
    J++;
    cout << "i = " << i << endl;
    cout << "J = " << J;
}
```

i = i+1

```
main()
{
    func();
    func();
    func();
}
```

gradeup
Prep Smart. Score Better.

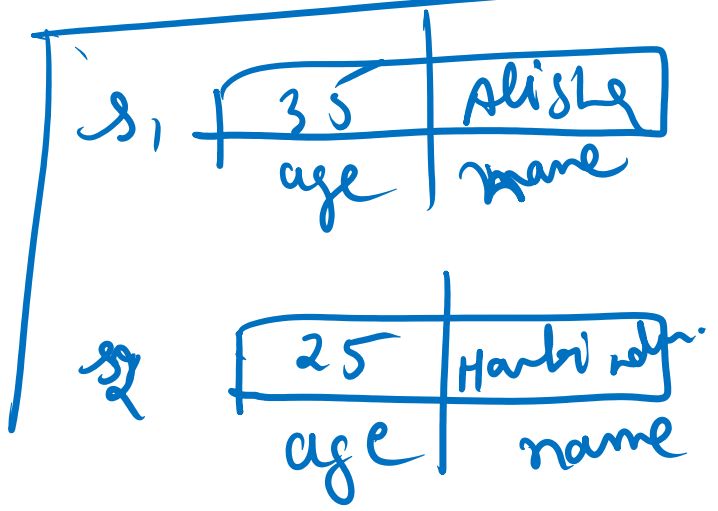
Call by value and call by reference in C++



gradeup
Prep Smart. Score Better.

C++ Object and Class Example

```
class student
{
public:
    int age;
    string name;
};
```




```
main()
{
    student S1, S2;
    S1.name = "Alisha";
    S1.age = 35;
    S2.name = "Harbinder";
    S2.age = 25;
    cout << S1.name
    cout << "Total Age = "
    cout << S1.age + S2.age;
}
```

Objects

o/p

Alisha
Total Age
= 60

gradeup
Prep Smart. Score Better.

Inheritance

(*) When one object acquires all the properties & behaviour of parent object i.e. known as inheritance.

(*) It provide code reusability.

(*) It is used to achieve runtime polymorphism.

gradeup
Prep Smart. Score Better.

Initialize and Display data through method

```
class student
{
public:
    int id;
    string name;
void insert(int i, string n)
{
    id = i;
    name = n;
}
void show()
{
    cout << "Name is:" << name << endl;
    cout << "ID is:" << id;
}
main()
{
    student s1;
    s1.insert(23, "Ram");
    s1.show();
}
```

data member

member function

o/p:

Name is: Ram
ID is: 23