

C++ Revision Tour

CLASS XII

OOPs Concepts

1. Object

2. Class

- Data Abstraction
- Encapsulation
- Modularity
- Inheritance
- Polymorphism

Procedural Programming Vs. Object Oriented Programming

Procedural Programming	Object Oriented Programming
No Real World Concept	Real World Concept
Emphasis on Algorithm rather than Data	Emphasis on Data rather than Algorithm
Based on Procedure	Based on Object
Change in Data Type propagate to change in all function/hole program	Does not
How to get concept	What to get concept
No reusability of code	Reusability of code
Simple to develop	Tricky to develop

C++ Basics

- Tokens
- Identifiers
- Literals
- Constants
- Punctuators
- Operators

Data Types

- Fundamental Data Types

- int
- char
- float
- Double
- void

- Derived Data Types

- Arrays
- Functions
- Pointers
- References

- User Defined Derived Data Types

- Class
- Structures
- Union
- Enumeration

- Data Type Modifiers

- Signed
- Unsigned
- Long
- Short

Flow of Control

- Sequential
- Selection
- Iteration

Console I/O Operation

- Single Character Functions
 - `getch()` `putch()`
 - `getchar()` `putchar()`
- String Functions
 - `gets()` `puts()`
 - `Cin.getline()` `cout.write()`

Function

- Function
- Function Prototype
 - Argument(s)
 - Default Argument(s)
`int compute(int n1, int n2, int n3=5);`
 - Constant Arguments(s)
`int compute(int n1, int n2, const int n3=5);`
- Call-by-Value
- Call-by-Reference

Recursion / Recursive Function

```
int factorial (int x)
{
    if(x<=1)
        return 1;
    else
        return x * factorial(x-1);
}
```

Scope Rules

- **auto**
 - eg. `int mark`
or
`auto int mark;`
- **register**
 - Same as 'auto' only difference it stored in CPU not in memory so it is fast.
 - eg. `register int mark;`
- **extern**
 - eg. `extern int mark;`
- **static**
 - eg. `static int mark;`

Storage Class Specifiers

Used for function

- static //file scope
- extern //out side file scope

Thanks.....

By :

DINESH PATEL
PGT [CS]
KV NAD KARANJA