Never heard that word!

1. Inline 🡪 Generally used as inline function. Normally when a function is called the control is transferred where the code is defined but in case of inline function the compiler while compilation the whole code is copied where the function is called.

An inline function is defined same as function just inline keyword before function name.

1. Abstract class, pure virtual functions (abstract functions) 🡪 Sometimes we don’t know about the full implementation of several functions declared in a class and they are reimplemented when the implementation is clear, such classes are called abstract class. While the pure virtual functions are to be overridden in derived class else the derived class also becomes abstract class.

These are declared as🡪

virtual float calculateVolume() = 0;

1. Abstract Data Types (ADT) 🡪 (Meaning of abstract is existing in thought or idea but lacking concrete existence) these are the data types that we don’t need to know the inner structure. E.g. In case of stack we use pop() and push() operations but we are totally unaware of the inner implementation. @2 a class we create with different variables and operations like insert delete display all combined represents a ADT.
2. Data abstraction 🡪 It is the concept of OOP , it refers to providing only essential data to the outside world and hiding the background details.
3. Memory leak 🡪 when we explicitly allocate a memory block and we don’t remove it after use it will lead to waste of memory called memory leak.
4. Tail recursion 🡪 if in a piece of code if recursion call is at last then it’s called tail recursion. In this case all the functions will be performed at calling time only and the function will not be performing at returning. Means at returning time it doesn’t have to perform anything at all. In case of tail recursion using loop is more efficient. (space)
5. Tree Recursion 🡪 If a function calls itself more than one time.
6. Hamming code 🡪 It’s a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors. It was developed by R.W. Hamming for error correction. [more](https://www.electrically4u.com/hamming-code-with-a-solved-problem/)
7. Implicit and explicit memory allocations
8. Auxiliary arrays 🡪 helper arrays normal arrays just created to help in the working.
9. C++ initializes strings empty else all other variable hold garbage value.
10. 🡪 this means dereference and access the thing.

Eg. student \*chan = new student();

(\*chan).marks;

Or

Chan ->marks;