

Agenda

- Function
- Function Overloading
- Name Mangling and Mangaled Name
- Default Argument Function
- Inline Function
- Revision
 - ctor, this pointer
- Menu Driven Code
- Enum
- Modular Apporach
- Dynamic Memory Allocation
 - new and delete operator

Function (demo01)

Function Overloading (demo02)

- Defining multiple functions with same name but differnt signature
- Differnt signature means ->
 - 1. If type of parameters are same then their count should be differnt
 - 2. If no of parameters (count) are same then their types shoulld be different
 - 3. If no of parameters and type of parameters are same then their sequence/order should be different
- At the time of function overloading return type of the function is not considered, as catching the value from the functions is optional
- Function overloading is an example of Compile Time Polymorphism

Name Mangling and Mangaled Name (demo02)

- The process that compiler follows to add extra information to the function at the time of function overloading is called as name mangling.
- the name that gets created after adding the extra information we call it as mangaled name.

Default Argument Function(demo03)

- A function that has default values assigned to its parameters is called as default argumnet function.
- For a default argument function the default values must be assigned from the rightmost parameter.

inline function (demo04)

- It is a request made towards the compiler to resolve the functions at compile time.
- The request can be accepted or rejected.
- Making functions inline can slow the speed of compilation however it will make the execution bit faster.
- It is recommended to use inline functions over the macros.

Revision

Menu Driven (demo06)

Enum (demo07)

Modular Apporach (demo08 and demo09)

Dynamic Memory Allocation (demo10)

-To allocate the memory dynamically use the operator new in cpp

```
int *ptr = new int;
```

- to assign the values at the dynamically allocated memory dereference the pointer.

```
*ptr = 10;  
cout<<"value = "<<*ptr<<endl;
```

- when dynamic memory allocation is done then we need to deallocate it manually, to deallocate it use the operator delete in cpp.

```
delete ptr;
```

- when poniter still points to the memory which is already deallocated, then such pointer is called as dangling pointer.
- to avoid the dangling pointer assign null to it after deallocation.

```
delete ptr;  
ptr = NULL;
```

- To check for memory leaks use the tool called as valgrind.
- to install use the below command
 - `sudo apt install valgrind`
- to use it use below command
 - `valgrind ./a.out`