Function

Mohammad Tasin

Agenda

- inline function
- default arguments
- reference variables
- Call by reference
- Call by value
- Call by address

inline function

- **Function in a program to save memory space which becomes appreciable when a function is likely to be called many times.**

- **Every time a function is called, It takes lot of extra time in executing.**

- **To eliminate the cost of call to small functions , C++ proposes a new feature called inline function.**
- **An inline function is a function that is expanded in a line when it is invoked.**
- **Compiler replaces the function call with the corresponding function code**
- **inline is a request not a command**

To the compiler may ignore the request in some situations :

Few of are them :-

- **Function containing loops, Switch or goto.**
- **Function with recursion.**
- **containing static variable.**

Syntax →

```
inline int add(int, int);
int add(int a, int b)
{
    return a + b;
}
```

default arguments

- We can set default values to the arguments in a function to let them allow to invoke it without passing value to the corresponding receiving variable.
- It is not necessary that all arguments should have some default values

- **Function cannot have non default argument after a default argument.**

  - ✓ **int add(int = 0, int =0, int =0);**
  - ✓ **int add(int, int =0, int =0);**
  - ✓ **int add(int, int, int =0);**

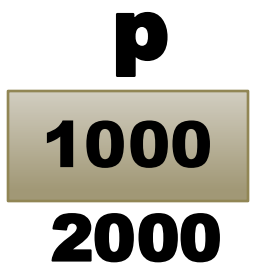
Reference variables

Address == Reference

```
int x =10;
```

ordinary variable

```
int *p;
p = &x;
```



Pointer variable

```
int &y=x;
y++;
```

Reference variable

Call by reference

```
#include<iostream>
using namespace std;
inline void add(int&,int&);
int main()
{
    int x =10, z =20;
    add(x,z);
    cout<<x<<endl<<z;
}
void add(int &a, int &b)
{
    a = 50;
    b = 100;
}
```

Call by value

```
#include<iostream>
using namespace std;
int Add(int, int);
int main()      {
    int x = 10, z = 20;
    cout<<"Sum is "<<Add(x, z)<<endl;
    return 0;
}
int Add(int a, int b)     {
   return a+b;
}
```

Call by address

```
#include<iostream>
using namespace std;
int Sum(int*, int*);
int main()
{
    int x = 10, z = 20;
    cout<<"Sum is "<<Sum(&x,&z)<<endl;
    return 0;
}

int Sum(int *a, int *b)
{
   return *a+*b;
}
```