

Void

{ swap (int a, int b)

int temp = a;

a = b;

b = temp;

}

int main()

{

int a = 10, b = 20;

swap (a, b);

cout << a << b << endl; //10 //20

}

Call by pointer.

void swap(int *a, int *b) {

int temp = *a;

*a = *b;

*b = temp;

}

int main() {

int a = 10, b = 20;

swap (&a, &b);

cout << a << b << endl //20 //10

}.

→ Example

```
struct employee;
void print (float value);
void print (Employee value);
print ('a');
```

// method with print (float)

→ ①

```
void f1 (int, char); ①
void f1 (char, int); ②
void f1 (short, char); ③
void f1 (float, int); ④
void f1 (char, bool); ⑤
```

f1('A', 5.6);

Exact match → 3

Promotion → 2

Conv → 1

{ any decimal no you see
is double
if after decimal no you
type 3.5f is float. }

so

Ans is (25) match

so it will through an error
(Ambiguous error).

now if f1('A', 'A');

(1,2) match

Ambiguous Error.

now if f1 ('A', TRUE);

⑤ Exact match

now if f1 (5.2, 6.2)

Inline function (to speed up our program)

- C++ provides an inline functions to reduce the function call overhead
- when the inline function is called whole code of the inline function gets inserted or substituted at the point of inline function call.
- This substitution is performed by the C++ compiler at compile time
- Inline function may increase efficiency if it is small.
- functions that are very short, say one or two statements are candidates to be inlined.

Inline function Example

```
#include <iostream>
using namespace std;
inline int add (int a, int b){
    return (a+b);
}
int main(){
    cout << "Addition of 'a' and 'b' is : " << add(2,3);
    return 0;
}
```

Output type cast or cast in C++.

Syntax : `a charVar = static_cast<char>(anIntVar);`



$$\begin{array}{r} 2 + \cancel{3} \\ \cancel{3} + 5 \\ \hline 2 + \cancel{5} + 2 \\ \hline \cancel{1} + 8 = 16 \end{array}$$

	M T W T F S S
Page No.:	YOUVA
Date:	(16)

- 1) static_cast<>() gives you a compile time checking ability,
C-style cast doesn't.

for Example

```
char c = 10;           // 1 bytes
int* p = (int*) &c;   // 4 bytes
* p = 5;              // run time error
int* q = static_cast<int*>(&c);
```

```
struct complex {
    int real;
    float image;
};

int main() {
    struct Complex C1;
    cout << "Enter C1 real: ";
    cin >> C1.real;
    cout << "Enter C1 image: ";
    cin >> C1.image;

    struct Complex C2;
    cout << "Enter C2 real: ";
    cin >> C2.real;
    cout << "Enter C2 image: ";
    cin >> C2.image;
```

```
struct complex {
    int real;
    float image;
};

int add();
int add();

int main() {
    struct complex C1 = {2, 3};
    struct complex C2;
    cin >> C2.real;
    cin >> C2.image;
    cout << "Addition of two complex no is: ";
    cout << add(C1, C2);
}

add()
{
    struct complex C3;
    C3.real = C1.real + C2.real;
    C3.image = C1.image + C2.image;
    return C3;
}
```

struct Complex add(struct Complex C1, struct Complex C2)

```
{
    struct complex C3;
    C3.real = C1.real + C2.real;
    C3.image = C1.image + C2.image;
    return C3;
}
```

```
cout << "Addition is " << C3;
((C1.real + C2.real) + (C1.image + C2.image));
Add();
struct complex C3 = add(C1, C2);
cout << C3.real << "+" << C3.image << "j";
```