User defined functions

functions

↳ 3 parts

1. function prototype or function declaration [upto us]
(inside the main)

2. function call (inside the main)
(compulsory part)

3. function definition (does, Tells steps to solve a problem)

Important pts

1) fn call — fn prototype (mandatory)

3) fn call — Always inside the main

2) fn definition is written outside main()

Above

fn prototype not reqd

Below

prototype is reqd

Team A

Team B

```
main ()
{
void swap(int, int);
swap(12, 36);   // Actual
}
void swap(int x, int y)
{
int temp;      -> 2
temp = x;      -> 1
x = y;         -> 1
y = temp;      -> 1
}
```

DT — void; fn name — swap; (int, int) -> 2 Arguments

Space = 6 bytes

Time = 4 units Time

```
main()
{
void swap(int, int);
swap(12, 36);
}
void swap(int x, int y)
{
x = x + y;   O(2)
y = x - y;   -> 2
x = x - y;   -> 2
}
```

Space = 4 bytes

Time = 6 units

Time = 6 units Time

Call by Value

→ Direct Values

→ if You make changes in formal arguments, it won't affect actual Arguments

→ Swapping

Call by Address

→ using pointer variable

→ Affect Actual Arguments

→ Swapping

Call by Reference

→ Reference Variable

→ Affects Actual Arguments

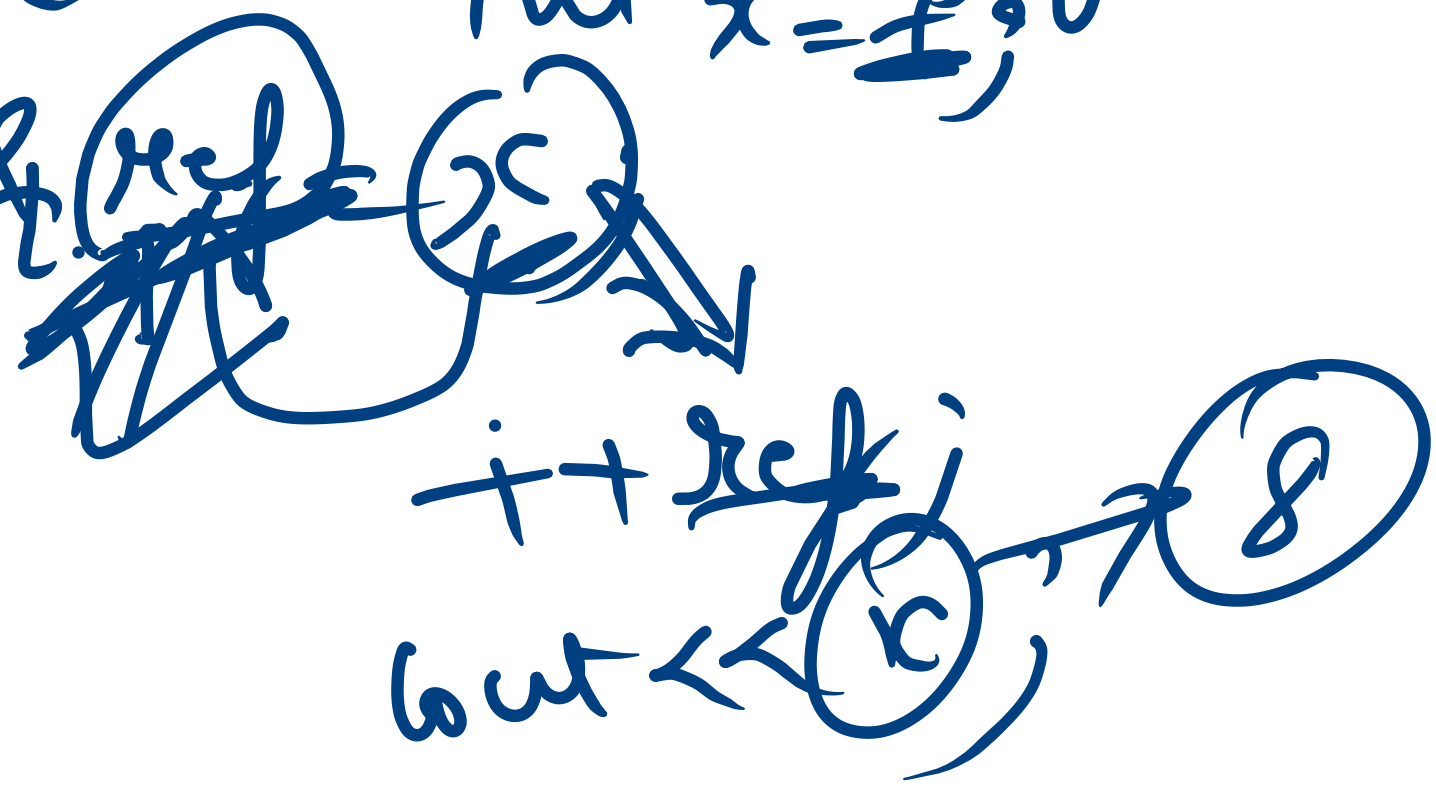
→ Swapping

Reference Variable

Alias (Another name) of a variable.

int x = 5;

int &ref = x;



Space & Time Complexity