

user defined functions

functions

↳ 3 parts

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

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

3. function definition (logic, Tells step by step how to solve a problem)

3) fn call

Always
inside
the main

Important pts

1) fn call fn defn (mandatory)

2) fn definition is written outside main()

Above

fn has to type
not reqd

Below

↓
Prototype is reqd

Question A
 main() fn name
 {
 void swap(int, int);
 }

swap(12, 36); // Actual.

space 6 bytes

```
void swap(int x, int y)
{
    int temp;
```

June - 2
4 yards
June.

```
int temp;
temp = x;
x = y;
y = temp;
```

main() Team b
{
void swap(int, int);
swap(12, 36);
}
void swap(int x, int y)

Бразил

Time = 6 weeks

$x = x + y$; $O(2)$
 $y = x - y$; $\rightarrow 2$
 $x = x - y$; $\rightarrow 2$

Time = $\frac{64 \text{ units}}{\text{hour}}$

call by value

- Direct values
- if you make changes in formal arguments, it won't affect actual arguments
- Swap

call by Address

- using pointer variable
- Affect Actual Arguments
- Swap

call by Reference

- Reference variable
- Affects Actual Arguments
- Swap

Reference Variable

Alias (another name) of a variable.

`int x = 1;`

`int &ref = x;`

A diagram illustrating the relationship between a reference variable and the variable it points to. The code `int &ref = x;` is shown. The variable `ref` is circled, and an arrow points from it to the variable `x`, which is also circled. The original code `int &ref = x;` is crossed out with a large 'X'.

`i++ ref j`

A diagram illustrating the relationship between a reference variable and the variable it points to. The code `int &ref = x;` is shown. The variable `ref` is circled, and an arrow points from it to the variable `x`, which is also circled. The original code `int &ref = x;` is crossed out with a large 'X'.

`cout << x;`

Space & Time Complexity

↓

Parameters