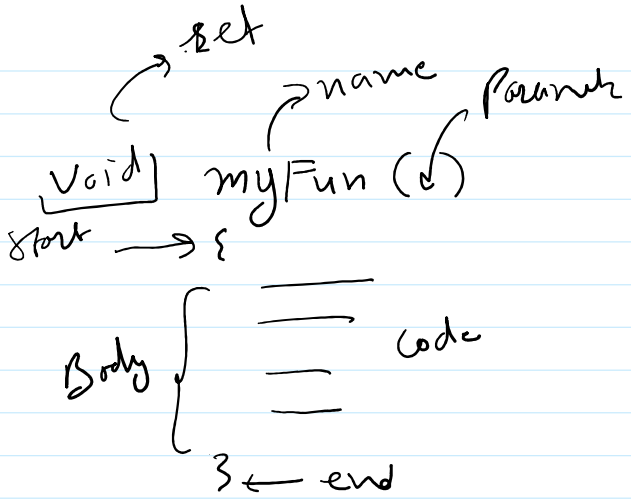
Functions

22 August 2023 11:54

Unit of code - reuse

Signature of a Function

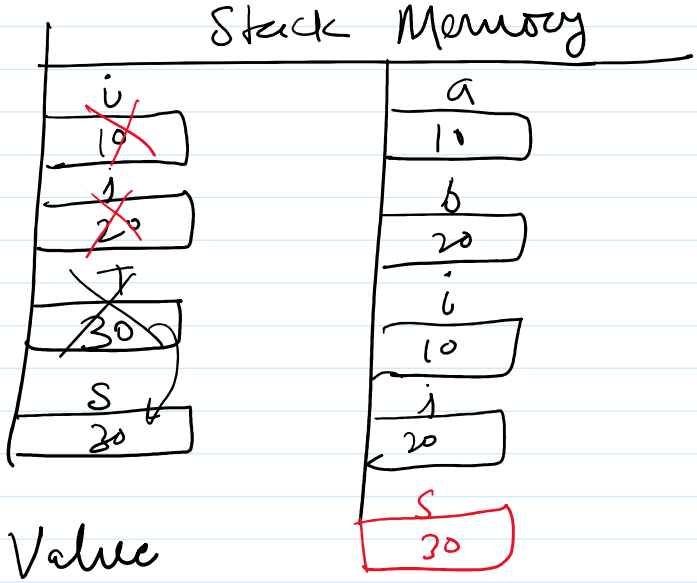
- name
- No. of parameter
- Type of parameter
- Return type




```
float sum (int i, int j)
{
}
```

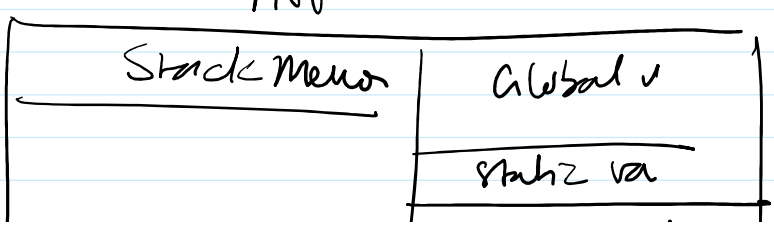
```
int DSum (int i, int j)
{
    i *= 2;
    j *= 2;
    return i+j;
}

main ()
{
    int a{10}, b{20}
    int s=DSum( a, b);
}
```

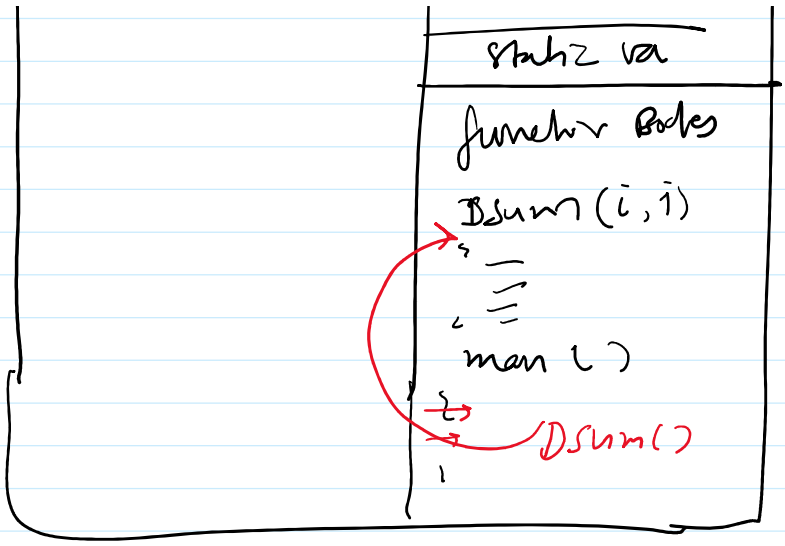
```
main ()
{
    sum (10, 20);   30
    int s = sum (10, 20);   30
}
```



Passing by Value

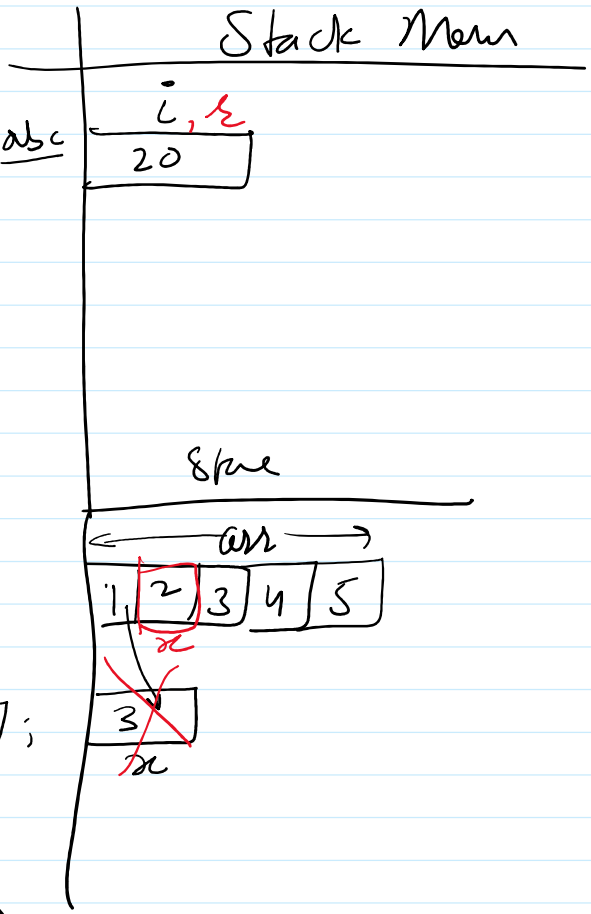
App Memory





References

A reference is an alias / another name of a variable.



→ int i {20};

int& r {i}, int& r = i;

(1) Range based for loop

int arr[] {1, 2, 3, 4, 5};

int i=0

for (auto& x : arr)

{ x+=2;

cout << x << '_' << arr[i++];

}

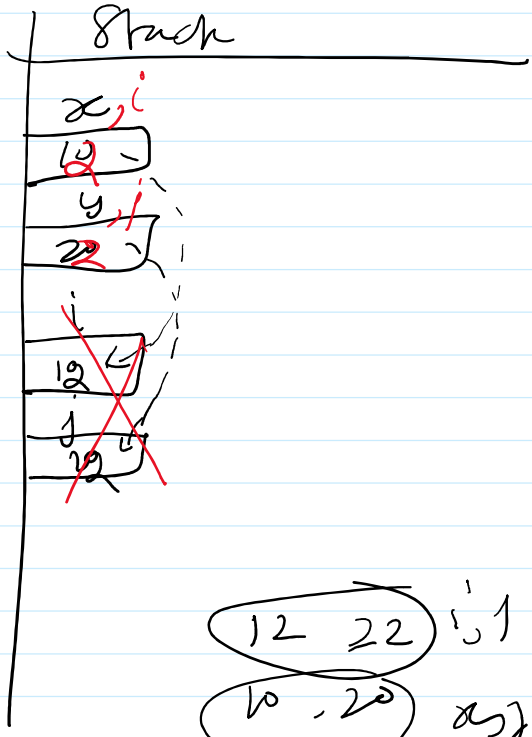
3,)

(2) function params → Pass by Ref

f(int& i int& i)

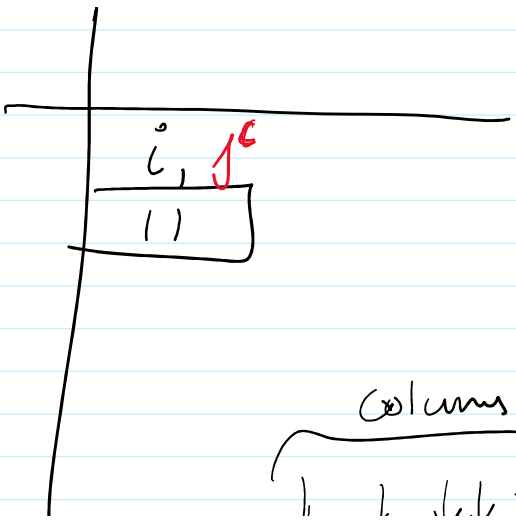
Stack

```
f1(int& i, int& j)
{
    i+=2;
    j+=2
    cout << i << ' ' << j;
}
f2()
{
    int x{10}, y{20};
    f1(x, y)
    cout << x << ' ' << y;
}
```




```
f()
{
    int x{10};
    x = DoTenTimes(x)
}
int DoTenTimes(x)
{
    return x*10;
}
```

```
{
    int i;
    int& j{i};
    int i{10};
    const int& j{i};
    i++
    cout << j;
}
```



cout << j;

const int& i const int& j

f(int i , int j)

{

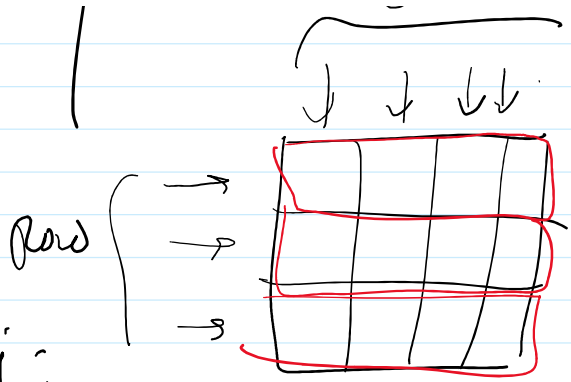
cout << i << " " << j;

}

f(int& i , int& j)

{

}



a[3][4]

a[i] → Row index