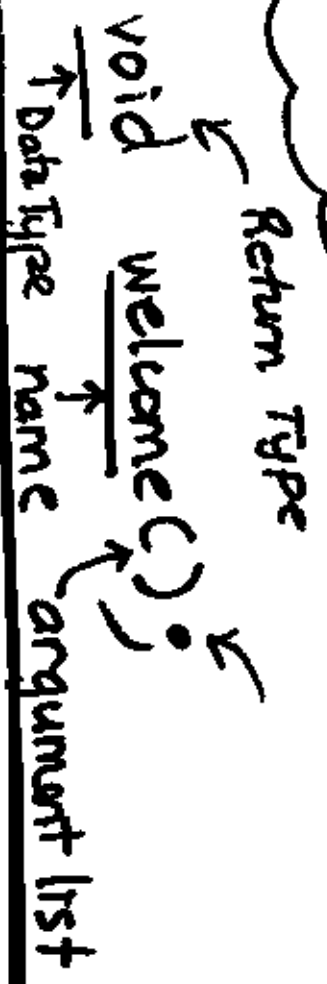


# Lecture 4



- Function Declaration
- Prototype

```
int main()
{
    welcome();
    return 0;
}
```

- Function call

---

```
void welcome()
{
    cout << "Message.....";
}
}
```

- Function Implementation
- Definition

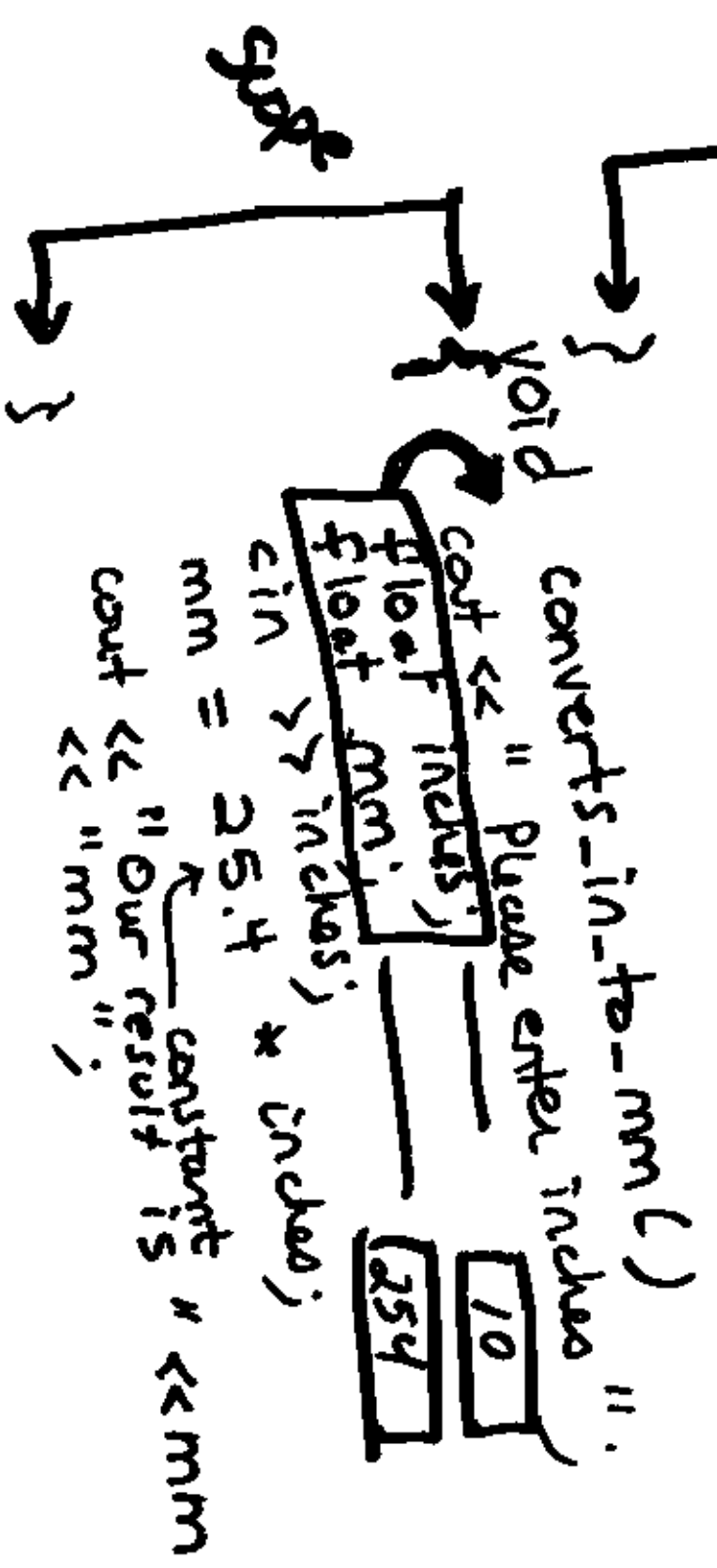
# Local Variables

~~int size; // global~~

```

int main()
{
  float mm;
  converts_in_to_mm();
  cin >> size;
}

```



OR float convert (float inches);

float convert (float);  
Data type of argument      Prototype

10.0

float inches, mm;

cin >> inches;

mm = convert (inches);      an

gave "in" value

call

cout << convert (10.0);

10.0

Local Variable  
with an initial  
value

float convert (float in)  
{  
  return 25.4 \* in;  
}

Definition

Pass by Value  
Call by Value

void convert(float, float &);

or  
void convert(float inches, float & mm); Prototype

Pass by value Pass by Reference

float inches, mm;

cin >> inches;

convert(inches, mm);

10.0 inches

254 mm

Call

void convert(float inches, float & mm);

mm = 25.4 \* inches;

10.0

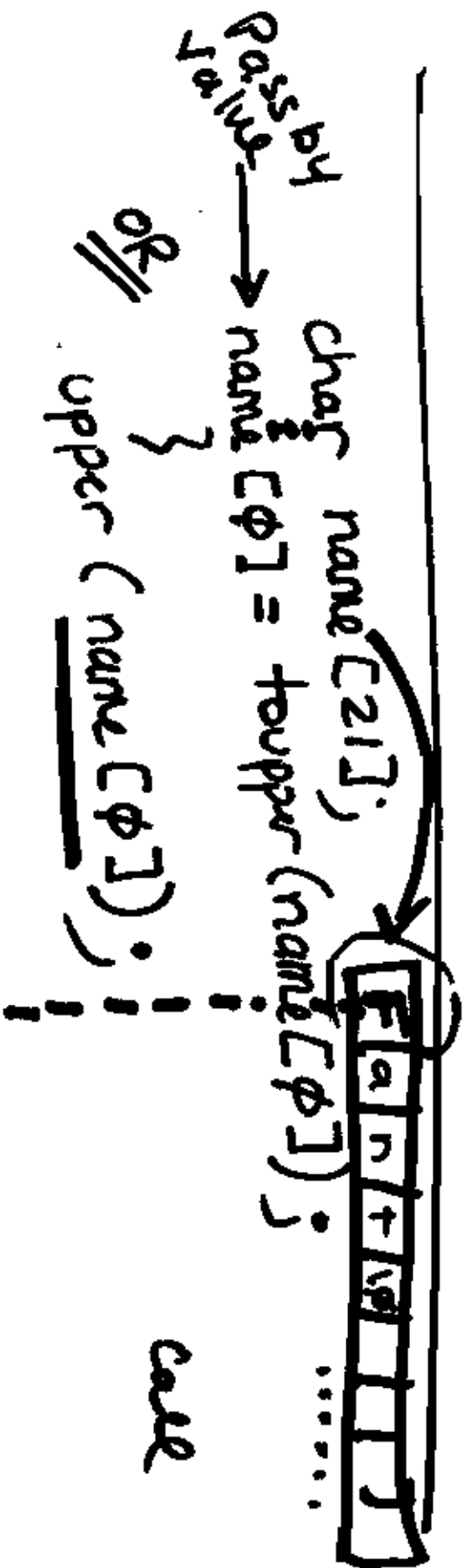
Definition

}

# Upper Case a Character

void upper (char & change);

Prototype



void upper (char & ch)

{

    ch = toupper (ch);

}

Definition

Pass by Reference

# Capitalize Every Character in the Array

```

void capitalize (char name [2]);
or
void capitalize (char [2]);

```

Prototpyze

char name [2];



capitalize (name);

! of the starting address  
the name array

void capitalize (char name [2])

{ for (int i=0; name[i]; ++i)

upper (name[i]);

// or: name[i] = toupper (name[i]);

}

Defination

(A)

---

```
for (int i = 0; name[i] != '\0'; ++i)
```

(B)

Library: `cstring`

```
int length;
length = strlen(name);
for (int i = 0; i < length; ++i)
```

← 4

---

```
for (int i = 0; i < strlen(name); ++i)
```

No  
(C)

```
#include <iostream>
using namespace std;
#include <cstring>
```

```
int main()
{
    int len = strlen(name);
```

```
char name1 [21];
char name2 [21];
```

```
int strlen(char c[]);
int strcmp(char c1, char c2);
int strcpy(char c1, char c2);
```

```
if (strcmp(name1, name2) == 0) // Same
    < < < // name1 < name2
    > > > // name1 > name2
```