WHAT IS A FUNCTION (IN SIMPLE TERMS)

CHAPTER 5

SURESH TECHS

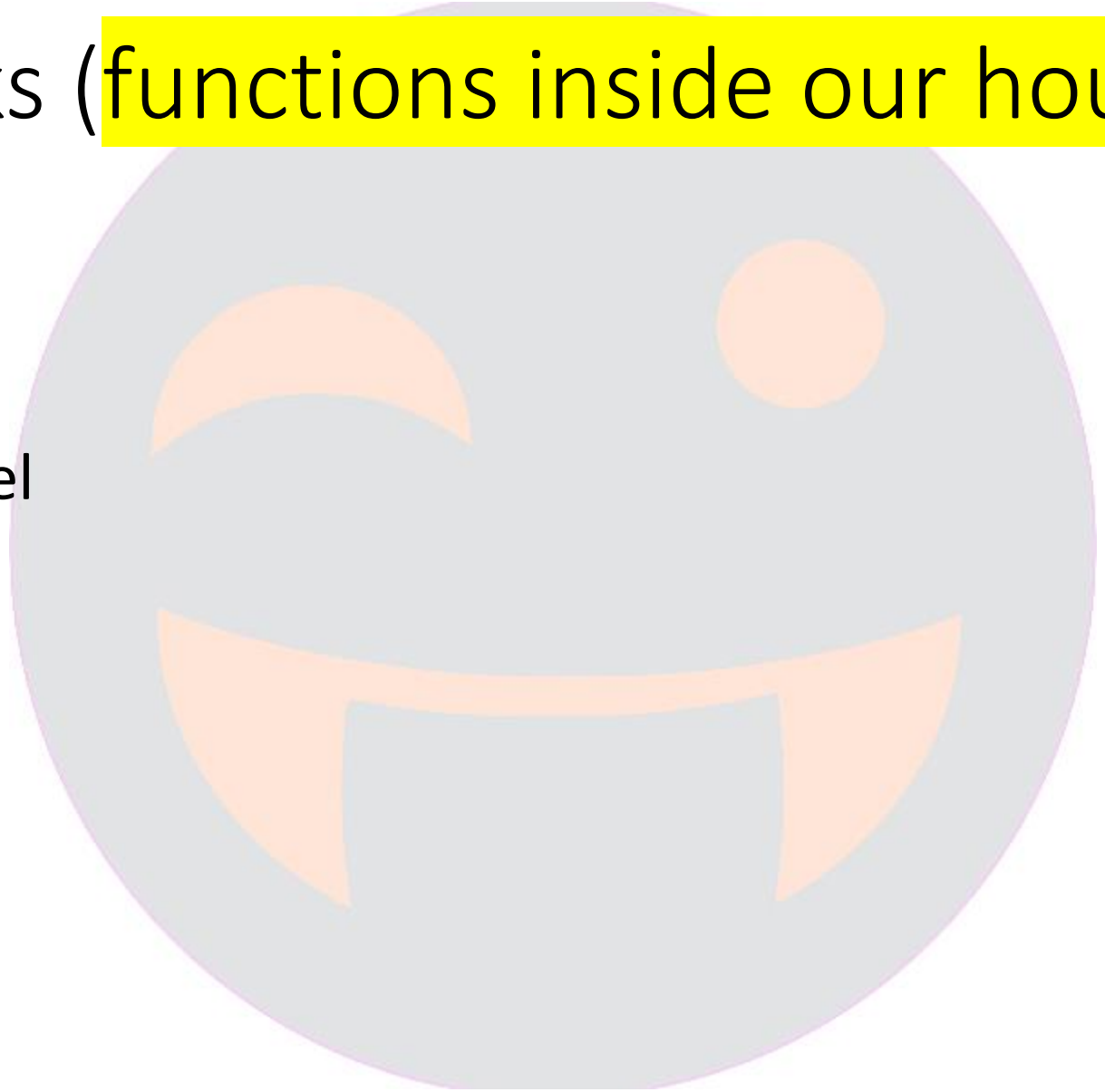
C PROGRAMMING COURSE

Function - Used to perform a specific task

- Also called method

Some tasks (functions inside our house)

- switchOnLight
- getWater
- changeChannel
- openLaptop



Statement;

```c
#include <stdio.h>
int main() {

    int number1, number2, sum;

    printf("Enter two numbers: ");
    scanf("%d %d", &number1, &number2);

    sum = number1 + number2;

    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
```

Change channel - task

1. **Take remote.**
2. **Find numbers.**
3. **Press specific channel number.**

- **Performs a piece of programming action.**
- It must be terminated by a semi-colon (;) (like an English sentence ends with .)
- Every statement must end with semi-colon ;

Function?

- A function is a set of statements that can optionally take **inputs**, do some specific **computation** and may produce **output**

- switchOnLight
- getWater
- changeChannel
- openLaptop

switchOnLight

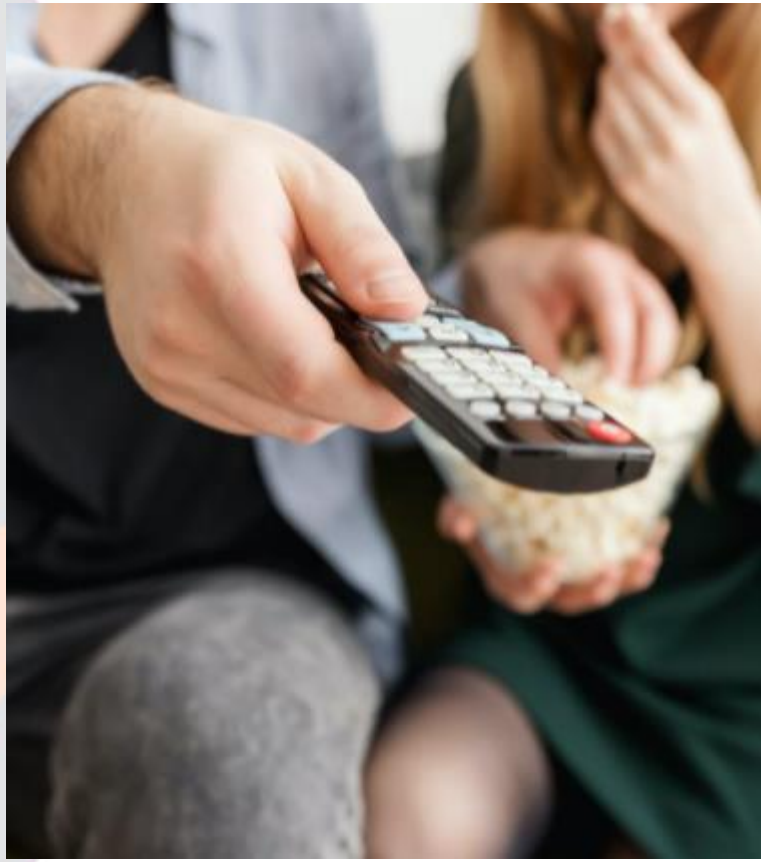
- Input: 2nd light
- Computation(statements): walk till switch board, raise hand and push the switch
- Output: light on

getWater

- Input: 1 glass of water
- Computation(statements): go to kitchen, take a glass, fill water and get
- Output: water/no water

changeChannel

- input: 21 channel
- Computation(statements): take remote, press 2 and 1 buttons
- Output: no output



openLaptop

- Input: no input
- Computation(statements): Hold laptop, open lid
- Output: no output

How to write a **function** in programming

- **switchOnLight**
- Input: 2nd light
- Computation(statements): walk till switch board, raise hand and push the switch
- Output: light on

```
char switchOnLight( number ){
  //statements
  return lighton
}
```

block

A block is a group of programming statements enclosed by braces { }, no need to put semicolon after the closing brace

How to write a function in programming

- **getWater**
- Input: 1 glass of water
- Computation(statements): go to kitchen, take a glass, fill water and get
- Output: water/no water

```
char getWater(number){
//statements
return water
}
```

How to write a function in programming

- **changeChannel**
- input: 21 channel
- Computation(statements): take remote, press 2 and 1 buttons
- Output: no output

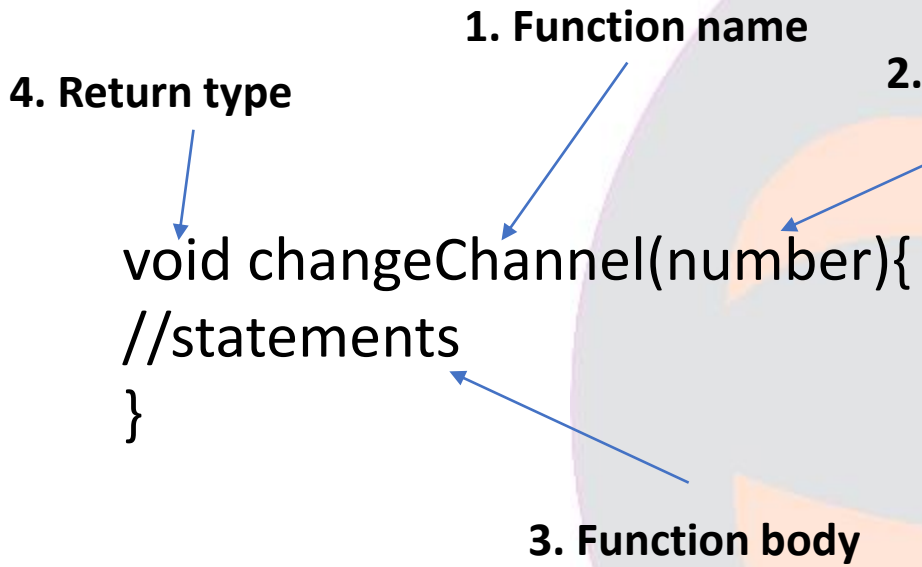
```
void changeChannel(number){
//statements
}
```

How to write a function in programming

- **openLaptop**
- Input: no input
- Computation(statements): Hold laptop, open lid
- Output: no output

```
void openLaptop(){
//statements
}
```

Each function will have these




```
char getWater(number){
//statements
return water
}
```

```
void openLaptop(){
//statements
}
```

How to run/call a function?

- switchOnLight
- getWater
- changeChannel
- openLaptop

```
char switchOnLight(number){
    //statements
    return lighton
}

char getWater(number){
    //statements
    return water
}

void changeChannel(number){
    //statements
}

void openLaptop(){
    //statements
}
```

- switchOnLight(2);
- getWater(3);
- changeChannel(22);
- openLaptop();

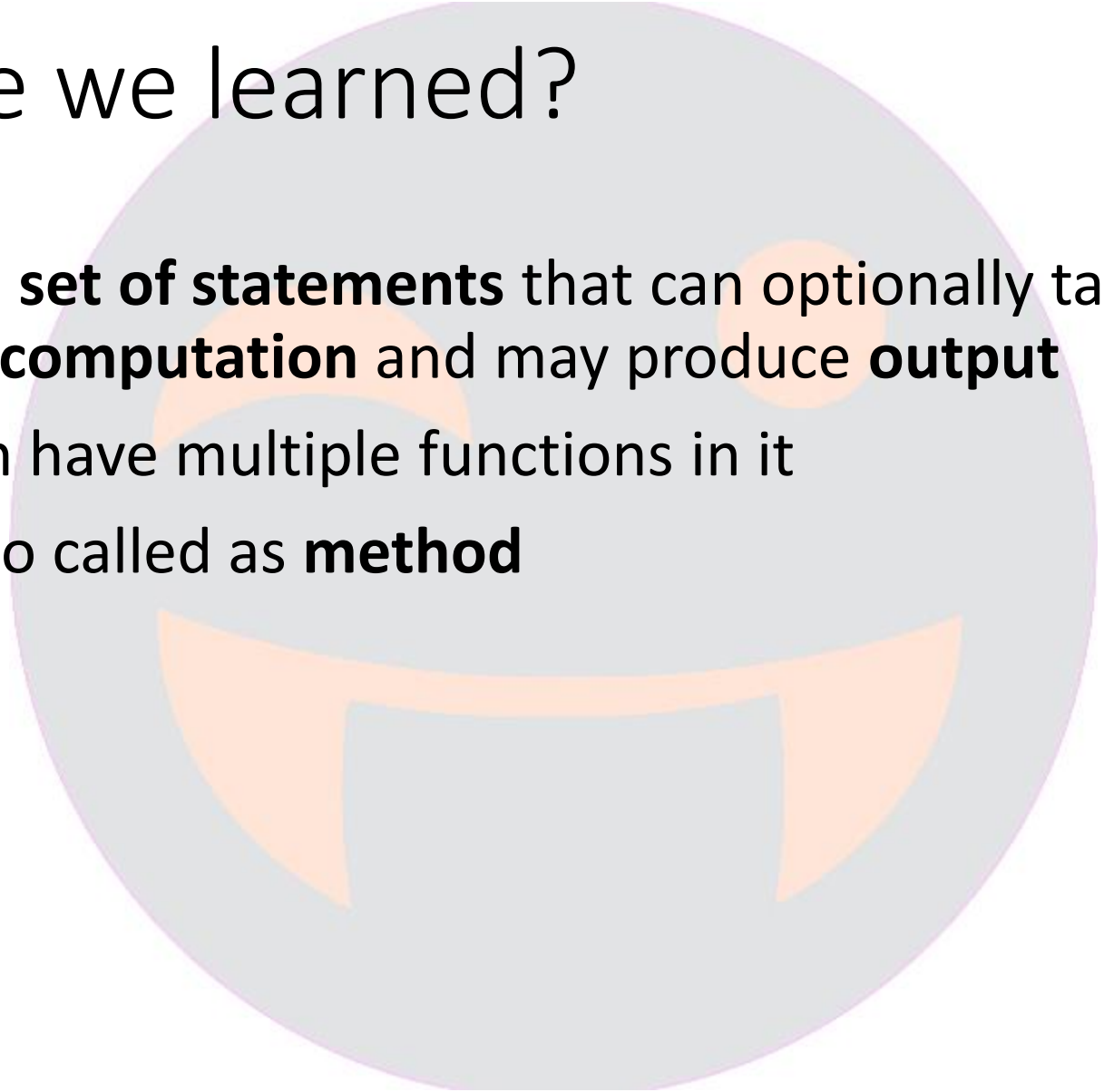
Can a program have multiple functions?

- YES absolutely
- A big task can be completed by making small tasks
- Ex: house.c

```
char switchOnLight(number){
    //statements
    return lighton
}

char getWater(number){
    //statements
    return water
}

void changeChannel(number){
    //statements
}

void openLaptop(){
    //statements
}
```

What have we learned?

- A function is a **set of statements** that can optionally take inputs, do some specific **computation** and may produce **output**
- A program can have multiple functions in it
- Function is also called as **method**



What next?

- Let’s write our first program