SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 23CST101 PROBLEM SOLVING AND C PROGRAMMING

I YEAR/ V SEMESTER

UNIT – IV FUNCTIONS AND POINTERS

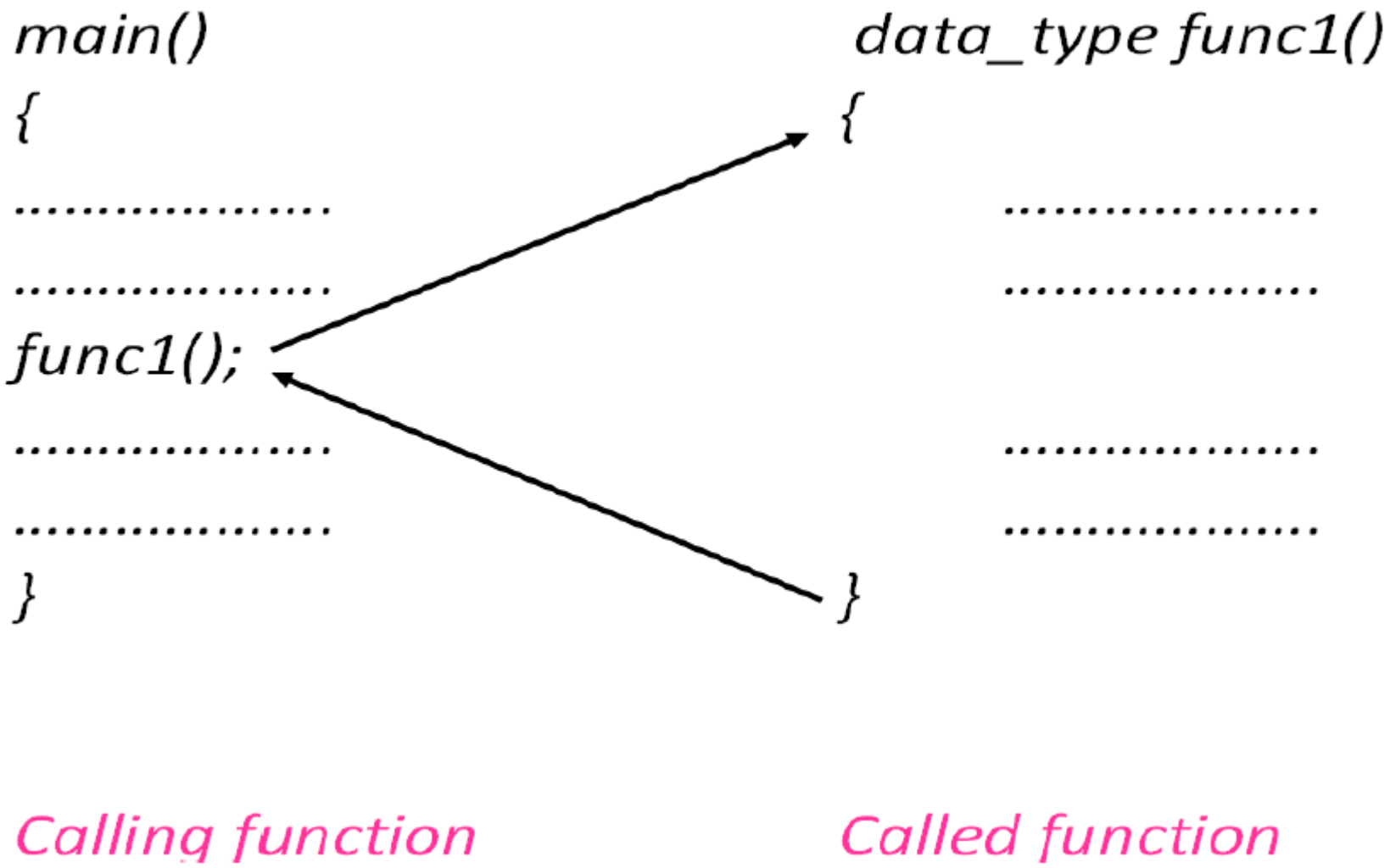
User Defined Function

Dr.B.Vinodhini

Associate Professor

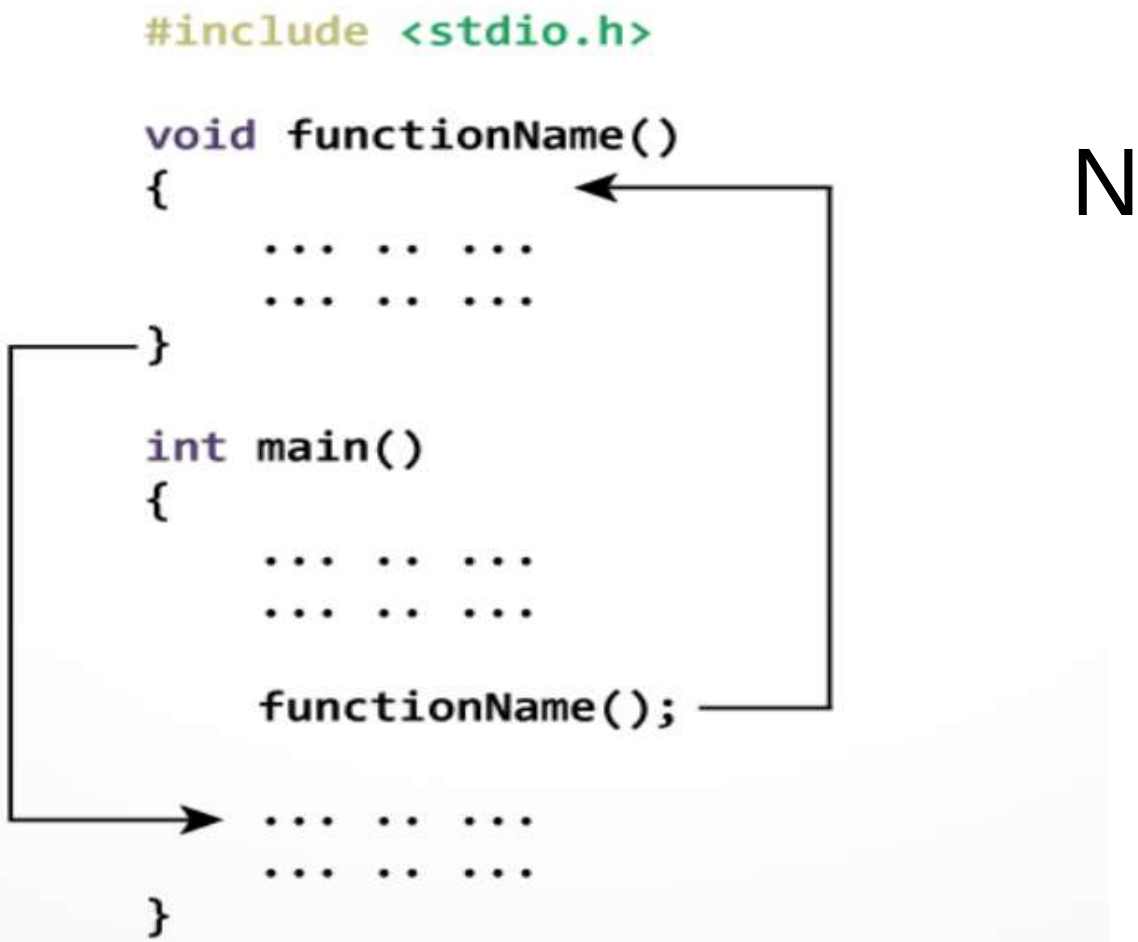
Department of Computer Science and Engineering

USER DEFINED FUNCTIONS



Functions

How function works in C programming?



Note, function names are identifiers and should be unique.

```c
#include <stdio.h>

int areaOfRect(int length, int breadth)
{
    int area;
    area = length * breadth;
    return area;
}
int main()
{
    int l=10, b=5;
    int area = areaOfRect(l, b);
    printf("%d\n", area);

}
```

```c
int add(int, int);

int main()
{
    int m = 20, n = 30, sum;
    sum = add(m, n);
    printf("sum is %d", sum);
}

int add(int a, int b)
{
    return ( a + b );
}
```

This is the way you call a function.

NOTE: while calling a function, you should not mention the return type of the function

Also you should not mention the data types of the arguments

This is the way how you define a function.

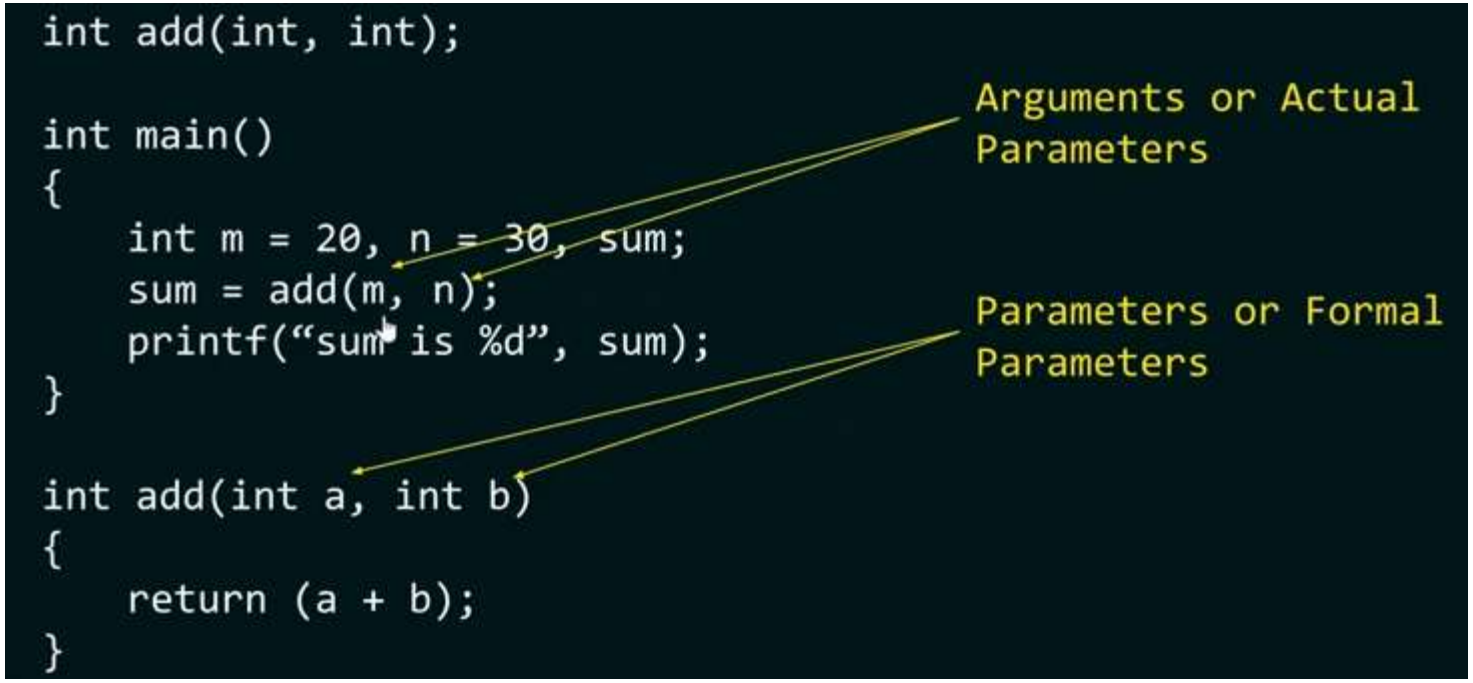
NOTE: it is important to mention both data type and name of parameters

```c
int add(int, int);

int main()
{
    int m = 20, n = 30, sum;
    sum =  50
    printf("sum is %d", 50 );
}

int add(int a, int b)
{
    return (   50   );
}
```

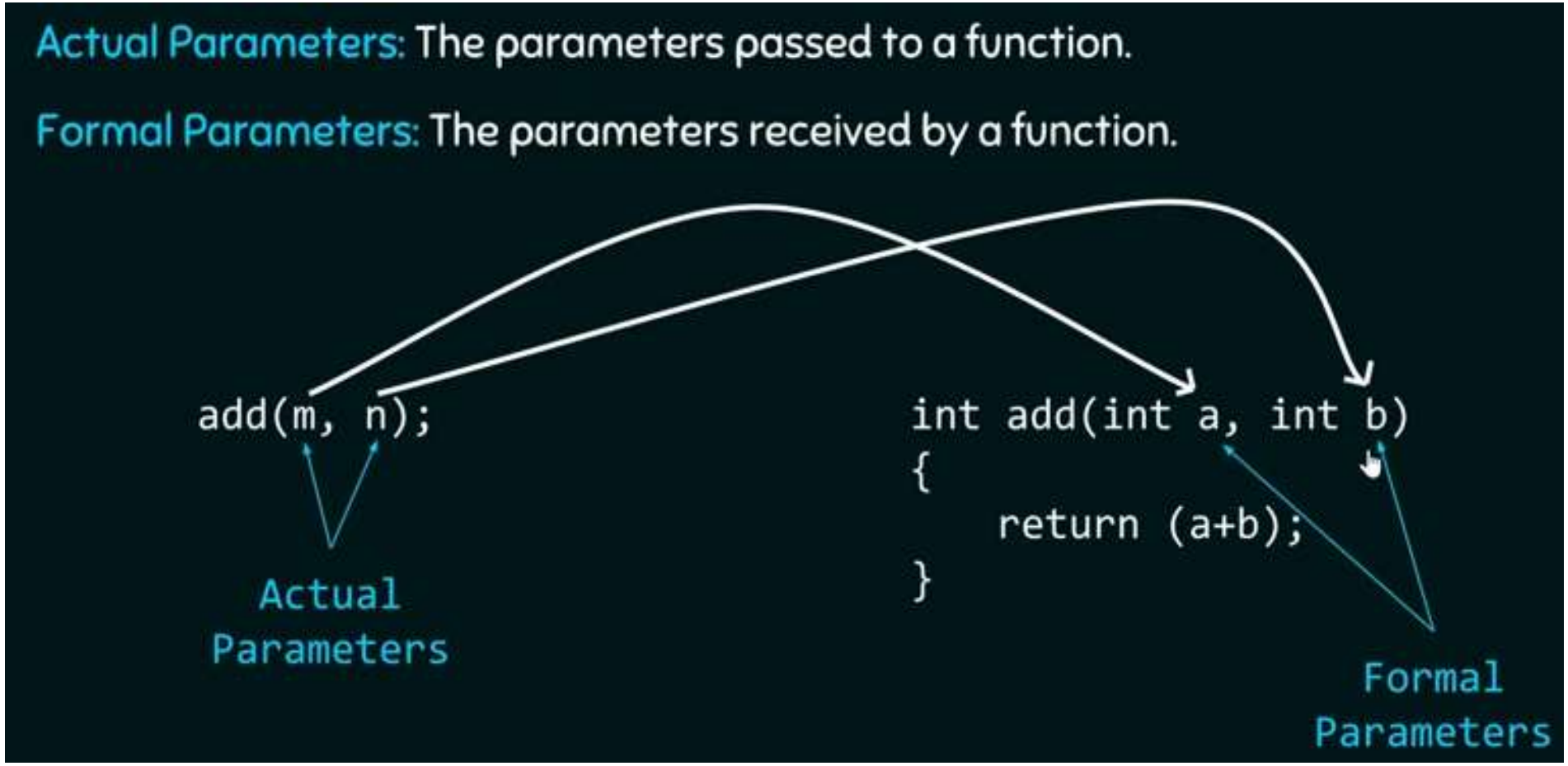
Parameter is a variable in the declaration and definition of the Function

Argument is the actual value of the parameter that gets passed to the function

Note :Parameter is also called as Formal Parameter and Argument is also called as Actual Parameter



- The type of arguments passed to a function and the formal parameters must match, otherwise, the compiler will throw an error.
- A function can also be called without passing an argument.


Actual Parameters: The parameters passed to a function.
Formal Parameters: The parameters received by a function.
add(m, n);
int add(int a, int b)
{
return (a+b);
}
Actual Parameters
Formal Parameters

FUNCTION PROTOTYPE(fn Name, args,return value)

Prototype helps to check the arguments & return value of the function

Types of Function

1.Without arguments and return value

2.Without arguments and with return value

3.With arguments and no return value

4.With arguments and with return value

Without arguments and without return value

```
Void main()
{
      void add(void);
      add();
      getch();
}
Void add()
{
      int a,b,c;
      printf(“Enter a &b”);
      scanf(“%d %d”,&a, &b);
      c=a+b;
      printf(“%d”,c);
}
```

Without arguments and with return value

```
Void main()
{
      int c;
      c=add();
      printf(“%d”,c)
      getch();
}

int add()
{
      int a=5,b=5;
      return(a+b);
}
```

with arguments and no return value

```
void main()
{
        int a=6,b=7;
        add(a,b);
        getch();
}

add(int i,int j)
{
        int k;
        k=i+j;
        printf("%d",k);
}
```

With arguments and with return value

```
void main()
{
        int a=6,b=7,c;
        c=add(a,b);
        printf("%d",c);
        getch();
}

int add(int i,int j)
{
        int k;
        k=i+j;
        return(k);
}
```


Thank
you