Reading input from the keyboard – scanf()

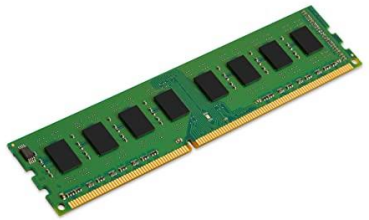
CHAPTER 23

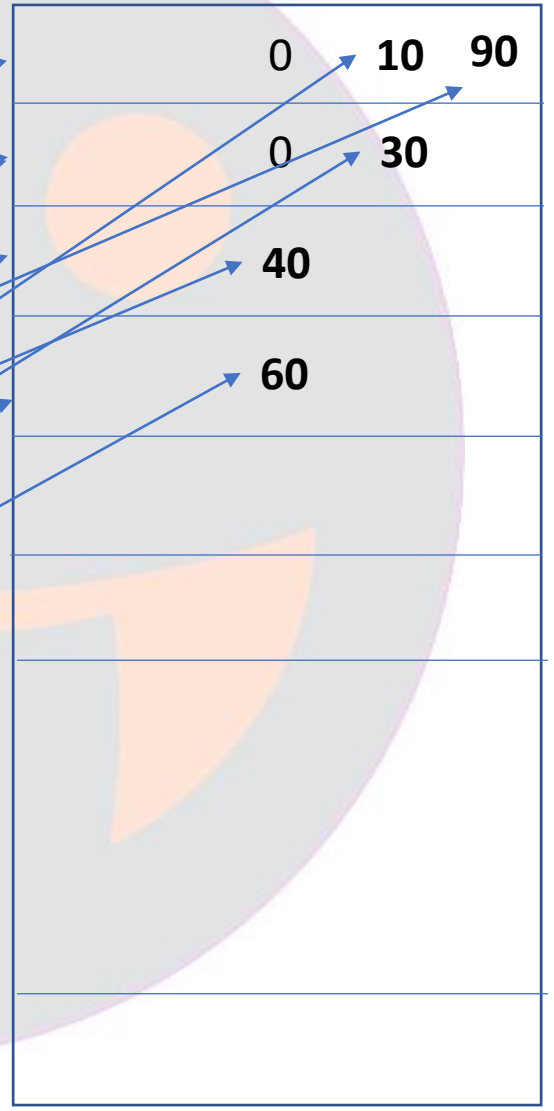
SURESH TECHS

C PROGRAMMING COURSE

scanf

```c
#include<stdio.h>
int main(){
    int a;
    int b;
    a = 10;
    b = 30;
    int sum = a+b;
    printf("%d\n",sum);
    printf("%d, %d\n",a,b);
    a = 90;
    int sub = a-b;
    printf("%d",sub);
}
```





00110100011010110

00110100011010111

00110100011010101

00110100011010001

Finding address of the variable

```c
#include<stdio.h>
int main(){
    int a = 10;
    int b = 20;
    printf("%d\n",a);
    printf("%d\n",&a);
    printf("%d",&b);
    return 0;
}
```

To get the address of the variable where the value is stored

scanf() - scanf("Format Specifier", Variable Address);

- Used to read data from the console

```c
#include<stdio.h>
int main(){
    int number;
    printf("Enter a number: ");
    scanf("%d",&number);
    int result = number * 20;
    printf("%d",result);
    return 0;
}
```

```c
#include<stdio.h>
int main(){
    int number = 5;
    int result = number*20;
    printf("%d",result);
    return 0;
}
```

The scanf() function reads the sequence of characters until it encounters whitespace (space, newline, tab)

int scanf(const char *format, Object *arg(s))

- Object: **Address** of the variable which will store the data
- char * : Contains the **format specifiers**

- **Format specifier:** It is a special character **used to specify the data type of the value being read**

- **Return value:**
  - If the function successfully reads the data, the **number of items read is returned**
  - In case of unsuccessful execution, a **negative number is returned**
  - If there is an input failure, **EOF** is returned

Why are we mentioning the address 🤔 ? - &a

```c
#include<stdio.h>
int main(){
    int number;
    printf("Enter a number: ");
    scanf("%d",&number);
    int result = number * 20;
    printf("%d",result);
    return 0;
}
```

- **This is done to store the value at the memory location of the variable**
- **stdio.h library must be included to use scanf()**

Read **name of your friend** 😍

```c
#include<stdio.h>
int main(){
    char name[10];
    printf("Enter your name: ");
    scanf("%s",&name);
    printf("%s",name);
    return 0;
}
```

But there is one speciality for char arrays

Note – **No need to provide address of** char **array**(string)

By default the variable itself points to the first address of char array(string) and therefore, there is <u>no need of adding an extra '&'</u>

```c
#include<stdio.h>
int main(){
    char name[10];
    printf("%p\n",&name[0]);
    printf("%p\n",name);
    printf("%p\n",&name);
    int n =10;
    printf("%p\n",n);
    printf("%p\n",&n);
    printf("Enter your name: ");
    scanf("%s",name);
    printf("%s",name);
    return 0;
}
```

Updating char array - string

```c
#include<stdio.h>
int main(){
    char name[10] = "suresh";
    //updating character value
    strcpy(name,"welcome");
    printf("%s",name);
    return 0;
}
```

Reading strings from terminal

```c
#include<stdio.h>
int main(){
    char name[10];
    printf("Enter name: ");
    /*scanf terminates it's input on the first white space it finds. A white space includes blanks, tabs, carriage returns, form feeds and new line*/
    scanf("%s",name);
    printf("Welcome %s",name);
    return 0;
}
```

Reading two integer values from terminal using space

- Same thing happens when you give string with a space

```c
#include<stdio.h>
int main(){
    char fName[10];
    char lName[10];
    printf("Enter name: ");
    /*scanf terminates it's input on the first white
space it finds. A white space includes blanks, tabs,
carriage returns, form feeds and new line*/
    scanf("%s %s",fName,lName);
    printf("Welcome %s %s",fName,lName);
    return 0;
}
```

Why is my full name not printing 🤔 🤔 ?

```c
#include<stdio.h>
int main(){
    char name[50];
    printf("Enter name: ");
    fgets(name, sizeof(name), stdin);//reading name
    printf("Name is: ");
    puts(name);//displaying name
    return 0;
}
```

Will talk more about strings later in a separate section

Write a program to print sum of two numbers - dynamically

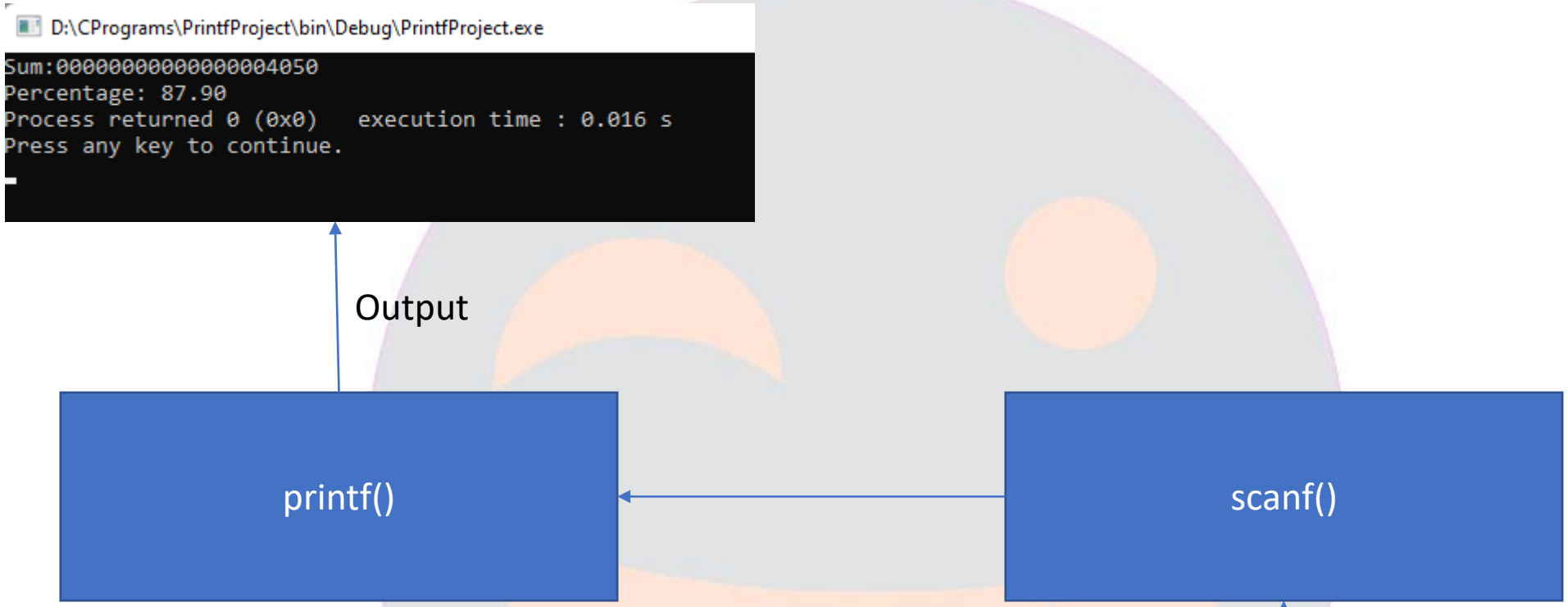
```c
#include <stdio.h>
int main() {

    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;
}
```



Managing input and output

What next?

Keywords

Identifiers

Strings

Special symbols

Constants

Operators

KISS CO