

printf() – in detail

# CHAPTER 22



**SURESH TECHS**

**C PROGRAMMING COURSE**

```
#include<stdio.h>
int main() {
    printf("Welcome to suresh techs");
    return 0;
}
```

- What if I want to print a number 🤔?
- printf = print formatted

**int printf (const char\* c-string, ...);**

```
#include<stdio.h>
int main() {
    printf(63540);
    return 0;
}
```

# int printf (const char\* c-string, ...);

- **Return value:** If the function successfully executes, it returns the total number of characters written to the output. If an error occurs, a negative number is returned

```
#include<stdio.h>
int main(){
    int a = printf("sureshtechs");
    printf("%d",a);
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int a = printf("%d %d \n",2,2);
    printf("%d",a);
    return 0;
}
```

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

# int printf (const char\* c-string, ...);

- **Arguments:** **c-string** is the string passed to the function. The string may contain **format specifiers**
- ... There can be additional arguments if there are format specifiers

```
#include<stdio.h>
int main() {
    int rank = 63540;
    printf("Rank is: %d", rank);
    return 0;
}
```

# Format specifiers

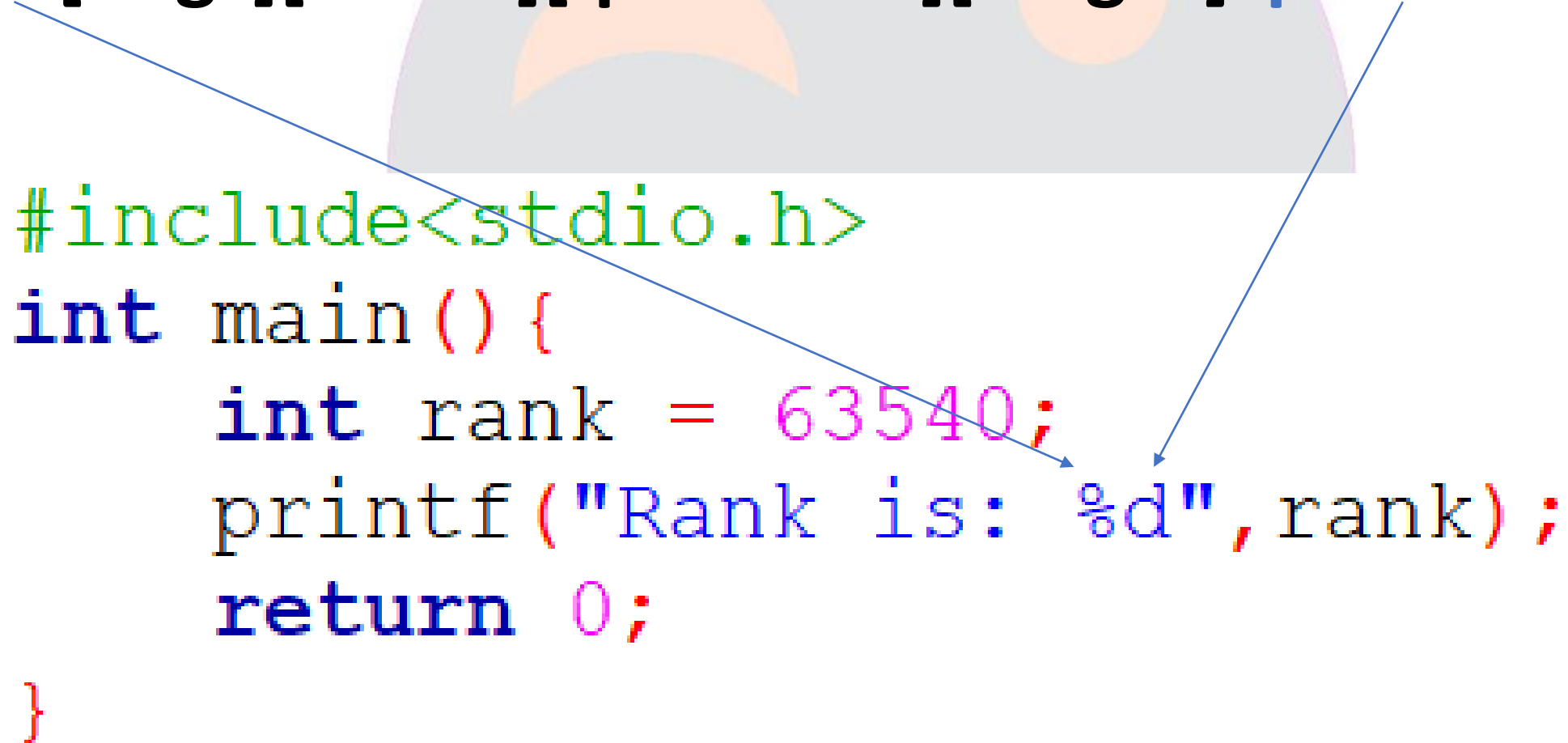
- Format specifiers **define the type of data to be printed** on standard output.
- We **need to use format specifiers** whether we are printing formatted output with **printf()** or accepting input with **scanf()**

SPECIFIER	USED FOR
<b>%c</b>	a single character
<b>%s</b>	a string
<b>%hi</b>	short (signed)
<b>%hu</b>	short (unsigned)
<b>%Lf</b>	long double
<b>%n</b>	prints nothing
<b>%d</b>	a decimal integer (assumes base 10)
<b>%i</b>	a decimal integer (detects the base automatically)
<b>%o</b>	an octal (base 8) integer
<b>%x</b>	a hexadecimal (base 16) integer
<b>%p</b>	an address (or pointer)
<b>%f</b>	a floating point number for floats
<b>%u</b>	int unsigned decimal
<b>%e</b>	a floating point number in scientific notation
<b>%E</b>	a floating point number in scientific notation
<b>%%</b>	the % symbol

# Format specifier

**%[flags][width][.precision][length]specifier**

```
#include<stdio.h>
int main() {
    int rank = 63540;
    printf("Rank is: %d", rank);
    return 0;
}
```



# %[flags][width][.precision][length]specifier

Specifier	Argument
d or i	Signed int
u	Unsigned int
o	Unsigned octal
x or X	Unsigned hexadecimal
g or G	Shortest representation
a or A	Hexadecimal floating point
c	Character
s	String of characters
p	Pointer address

```
#include<stdio.h>
int main(){
    int sum = 4050;
    float percent = 87.9;
    unsigned int age = 15;
    printf("sum: %d\n",sum);
    printf("percent: %f\n",percent);
    printf("age: %u",age);
    return 0;
}
```



# %[flags][width][.precision][length]specifier

Flag	Description
-	Left justify(default right)
+	Force print + sign with positives
Space	Add space, if no sign before value
#	Precedes o, x or X with 0
0	Left pad number with zeros

```
#include<stdio.h>
int main(){
    int sum = 4050;
    float percent = 87.9;
    unsigned int age = 15;
    printf("sum: %+d\n",sum);
    printf("percent: %f\n",percent);
    printf("age: %u",age);
    return 0;
}
```

# %[flags][width][.precision][length]specifier

Width	Description
Number	Minimum number of characters to print
*	Not specified in c-string, additional int value given

```
#include<stdio.h>
int main(){
    int sum = 4050;
    float percent = 87.9;
    unsigned int age = 15;
    printf("sum: %+7d\n",sum);
    printf("percent: %f\n",percent);
    printf("age: %u",age);
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int sum = 4050;
    float percent = 87.9;
    unsigned int age = 15;
    printf("sum: %+*d\n",10,sum);
    printf("percent: %f\n",percent);
    printf("age: %u",age);
    return 0;
}
```

# Don't worry

- These are very small things
- Modatlo telugu nerchukunnapu...

• అశ్వత్థేనుని పుత్రరత్నం తస్కస్కంబొట్లు 😐 😐

# %[flags][width][.precision][length]specifier

Width	Description
.number	Specifies number of digits to write
.*	Not specified in c-string, additional int value given

```
#include<stdio.h>
int main(){
    int sum = 4050;
    float percent = 87.9;
    unsigned int age = 15;
    printf("sum: %+*d\n", 10, sum);
    printf("percent: %.2f\n", percent);
    printf("age: %u", age);
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int sum = 4050;
    float percent = 87.9;
    unsigned int age = 15;
    printf("sum: %+*d\n", 10, sum);
    printf("percent: %.*f\n", 4, percent);
    printf("age: %u", age);
    return 0;
}
```

# What next?

- **scanf** in detail – to read from the user/keyboard

