

# C Programming: printf Format Specifiers

Dr. Amir

# Introduction

- ▶ The 'printf' function in C is used for formatted output (Kernighan and Ritchie, 1988).
- ▶ Format specifiers control how variables are printed.
- ▶ Common specifiers:
  - ▶ '%d' – Integer
  - ▶ '%f' – Floating-point
  - ▶ '%c' – Character
  - ▶ '%s' – String
  - ▶ '%o' – Octal
  - ▶ '%x' – Hexadecimal

# Integer Format Specifier: %d

- ▶ Used to print integers.
- ▶ Example:

```
#include <stdio.h>

int main() {
    int a = 10;
    printf("Integer: %d\n", a);
    return 0;
}
```

**Output:** Integer: 10

This example is adapted from ( Prata, 2004).

# Floating-Point Format Specifier: %f

- ▶ Used to print floating-point numbers.
- ▶ Example:

```
#include <stdio.h>

int main() {
    float b = 3.14159;
    printf("Float: %.2f\n", b);
    return 0;
}
```

**Output:** Float: 3.14

# Character Format Specifier: %c

- ▶ Used to print a single character.
- ▶ Example:

```
#include <stdio.h>

int main() {
    char c = 'A';
    printf("Character: %c\n", c);
    return 0;
}
```

**Output:** Character: A

# String Format Specifier: %s

- ▶ Used to print strings.
- ▶ Example:

```
#include <stdio.h>

int main() {
    char str[] = "Hello, Class!";
    printf("String: %s\n", str);
    return 0;
}
```

**Output:** String: Hello, Class!

# Octal Format Specifier: %o

- ▶ Used to print integers in octal (base 8) format.
- ▶ Example:

```
#include <stdio.h>

int main() {
    int a = 10;
    printf("Octal: %o\n", a);
    return 0;
}
```

**Output:** Octal: 12

# Hexadecimal Format Specifier: %x

- ▶ Used to print integers in hexadecimal (base 16) format.
- ▶ Example:

```
#include <stdio.h>

int main() {
    int a = 255;
    printf("Hexadecimal: %x\n", a);
    return 0;
}
```

**Output:** Hexadecimal: ff



# Step-by-Step Conversion: 255 to Hex

1. Divide 255 by 16:  $255 \div 16 = 15$  remainder 15
2. Convert remainder 15 to hexadecimal: **f**
3. Divide quotient 15 by 16:  $15 \div 16 = 0$  remainder 15
4. Convert remainder 15 to hexadecimal: **f**
5. Combine remainders in reverse order: **ff**

# Conclusion

- ▶ Format specifiers are crucial for formatted output in C.
- ▶ Use them correctly to display different data types (Kernighan and Ritchie, 1988; Knuth, 1973).
- ▶ Explore more specifiers like %u (unsigned integer) and %e (scientific notation).

# Unsigned Integer Format Specifier: %u

- ▶ Used to print unsigned integers (non-negative values).
- ▶ Example:

```
#include <stdio.h>

int main() {
    unsigned int a = 4294967295;
    printf("Unsigned Integer: %u\n", a);
    return 0;
}
```

**Output:** Unsigned Integer: 4294967295

This example illustrates how unsigned integers are represented (Prata, 2004).

# Scientific Notation Format Specifier: %e

- ▶ Used to print floating-point numbers in scientific notation (exponential format).
- ▶ Example:




```
#include <stdio.h>

int main() {
    double b = 1234567.89;
    printf("Scientific Notation: %e\n", b);
    return 0;
}
```

**Output:** Scientific Notation: 1.234568e+06

Scientific notation is commonly used for very large or very small numbers (Knuth, 1973).

# References I

-  Kernighan, Brian W. and Dennis M. Ritchie (1988). *The C Programming Language*. 2nd. Prentice Hall.
-  Knuth, Donald E. (1973). *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley.
-  Prata, Stephen (2004). *C Primer Plus*. 5th. Sams Publishing.