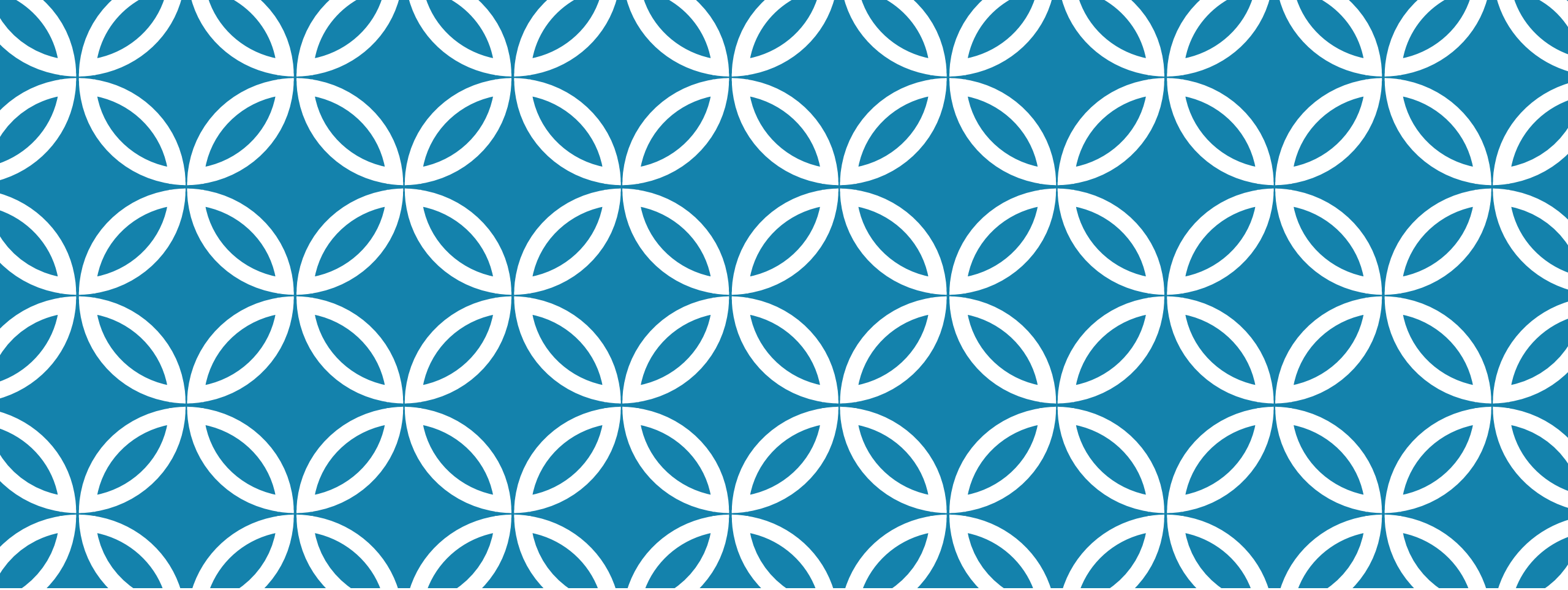

PAPER 102:
PROGRAMMING & PROBLEM SOLVING THROUGH C

Bidur Bhushan Handique
Dept. of Computer Science
St. Anthony's College

CONTENTS

- Console I/O functions
- Standard output library function: printf()
- Format Specification
- Optional Format Specification
- Printf(): Precision
- Programs

CONSOLE I/O FUNCTIONS

- The SCREEN and KEYBOARD are together known as CONSOLE
- Classified further as:
  - Formatted Console I/O Functions
  - Unformatted Console I/O Functions

The only difference between them is that the FORMATTED FUNCTIONS allow the input read from the keyboard or the output displayed on the VDU to be formatted as per the requirements.

- Eg:
  - Displaying Average and Percentage

CONTINUED

- Formatted Functions
  - Input: Scanf()
  - Output: Printf()
- Unformatted Functions
  - Input: getch(), getchar(), gets()
  - Output: putch(), putchar(), puts()

STANDARD OUTPUT LIBRARY FUNCTION: PRINTF()

General Form:

printf ("format string", list of variables) ;

Format String Contains

- Characters that are simply printed as they are
- Conversion specifications that begin with a % sign
- Escape sequences that begin with a \ sign

EXAMPLE

```
main( )
{
int avg = 346 ;
float per = 69.2 ;
printf ( "Average = %d\nPercentage = %f", avg, per ) ;
}
```

Output of the program:

Average = 346

Percentage = 69.200000

FORMAT SPECIFICATION

Commonly Used Conversion Characters for Data Output

Conversion Character	*Meaning*
`c`	data item is displayed as a single character
`d`	data item is displayed as a signed decimal integer
`e`	data item is displayed as a floating-point value with an exponent
`f`	data item is displayed as a floating-point value without an exponent
`g`	data item is displayed as a floating-point value using either `e`-type or `f`-type conversion, depending on value; trailing zeros, trailing decimal point will not be displayed
`i`	data item is displayed as a signed decimal integer
`o`	data item is displayed as an octal integer, without a leading zero
`s`	data item is displayed as a string
`u`	data item is displayed as an unsigned decimal integer
`x`	data item is displayed as a hexadecimal integer, without the leading `0x`

OPTIONAL FORMAT SPECIFIERS

- The field-width specifier tells **printf( )** how many columns on screen should be used while printing a value.

  - Eg:

    %10d

    print the variable as a decimal integer in a field of 10 columns

  Note:

  - If the value to be printed happens not to fill up the entire field, the value is right justified and is padded with blanks on the left.
  - If we include the minus sign in format specifier (as in **%-10d**), this means left justification is desired and the value will be padded with blanks on the right.

CONTINUED

```
#include <stdio.h>

main()          /* minimum field width specifications */

{
    int i = 12345;
    float x = 345.678;

    printf("%3d %5d %8d\n\n", i, i, i);
    printf("%3f %10f %13f\n\n", x, x, x);

}
```

Output

```
12345 12345    12345

345.678000 345.678000    345.678000
```

PRINTF(): PRECISION

1. The maximum number of decimal places for a floating point value , or for characters in a string can be specified.

2. A floating point number will be rounded if it must be shortened to conform to a precision specification.

CONTINUED

Example:

```
float x = 123.456;
printf("%7.3f  %7.1f", x, x);
```

O/P

```
123.456    123.5
```

EXAMPLE

```
#include<stdio.h>
int main()
{
float f_var=10.12576893;
int wid_prec;
printf("output 1: %f\n",f_var);
printf("output 2:%20f\n",f_var);
printf("output 3:%-20f\n", f_var);
printf("output 4:%+f\n", f_var);
scanf("%i",&wid_prec);
printf("output 5:%*f\n",wid_prec, f_var);
printf("output 6:%.5f\n", f_var);
return 0;
}
```

PROGRAMS

- WAP to calculate the area of a circle, where the radius is given as 3, and print the result on the screen.$(A = \pi r^{2})$
- WAP to calculate the Net salary for employees, where the basic is given and allowance and deductions are calculated as given below, finally display the net salary.

  DA is 45% of Basic pay

  HRA is 30% of Basic pay

  PF is 8% of Basic pay

STANDARD LIBRARY FUNCTION: SCANF()

General Form:

scanf("format string", list of addresses of variables);

Eg:

scanf("%d %f", &a, &b);

& is 'address of' operator

So any value received from keyboard are stored in the address of the corresponding variable.

FORMAT SPECIFICATION

Commonly Used Conversion Characters for Data Input

Conversion Character	*Meaning*
`c`	data item is a single character
`d`	data item is a decimal integer
`e`	data item is a floating-point value
`f`	data item is a floating-point value
`g`	data item is a floating-point value
`h`	data item is a short integer
`i`	data item is a decimal, hexadecimal or octal integer
`o`	data item is an octal integer
`s`	data item is a string followed by a whitespace character (the null character \0 will automatically be added at the end)
`u`	data item is an unsigned decimal integer
`x`	data item is a hexadecimal integer
`[...]`	data item is a string which may include whitespace characters

EXAMPLE

Program:

```
#include <stdio.h>
void main()
{
int a,b;
scanf("%2d %2d",&a,&b);
printf("a=%d b=%d",a,b);
}
```

Output:

```
1234
a=12 b=34

1 2
a=1 b=2
```

```
123 456
a=12 b=3

123456
a=12 b=34
```

USING MAXIMUM FIELD WIDTH:

1. It can be specified by preceding the conversion character by an unsigned integer.

2. The data item may be composed of fewer characters than the specified field width, but should not exceed the specified field width

3. Any characters that extend beyond the specified field width will not be read.

PROGRAM

1. Write a program to check if an integer input by the user is positive or negative.

2. Write a program to input three numbers and find out the

- biggest among them
- smallest among them

3. Write a program to input the Emp no., basic pay and designation code(101, 02) of an employee. Compute the DA-50% of BP, HRA is 40% of BP, PF-10% of BP. If designation code is 101 then give 2000 as bonus, for all others bonus is 1550. Calculate Gross and Net salary and display the pay slip in the form:

Emp no. Basic pay DA HRA PF Bonus GS NS