Parul® University

Fundamental of Programming:

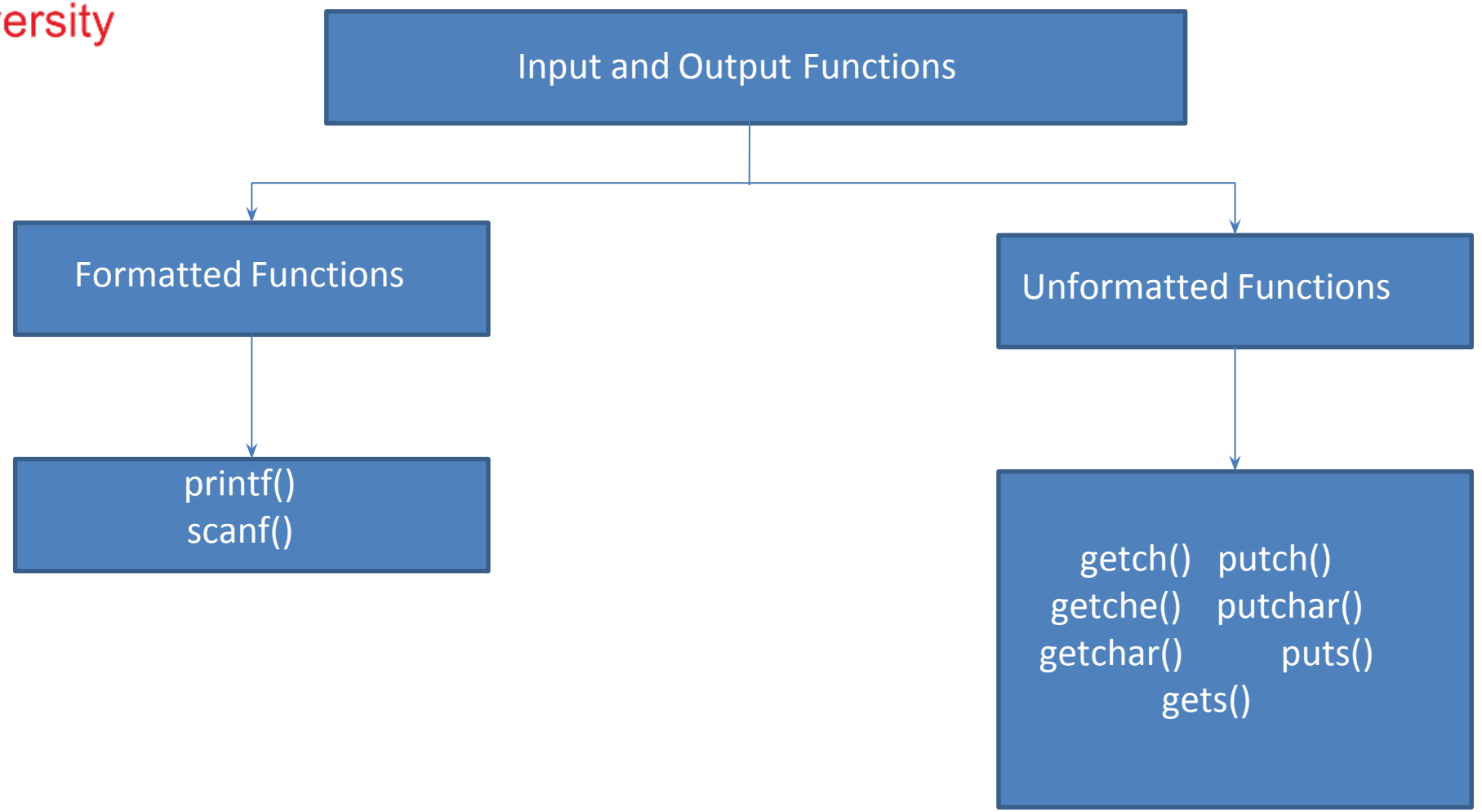
Prof. Sakaldeo Kumar Yadav, Assistant Professor

Computer Science & Engineering

Parul®
University

Managing input and output operations

Parul® University



Formatted Functions

- It read and write all types of data values.
- Require format string to produce formatted result
- Returns value after execution

Unformatted Functions

- Works only with character data type
- Do not require format conversion for formatting data type

Parul® University

printf() function

1. This function displays output with specified format
2. It requires format conversion symbol or format string and variables names to the print the data
3. The list of variables are specified in the printf() statement
4. The values of the variables are printed as the sequence mentioned in printf()
5. The format string symbol and variable name should be the same in number and type

```
void main()
{
      int Num1 = 2;
   float Num2=2.2;
   char LetterCh = 'C';

      printf("%d %f %c", Num1, Num2, LetterCh);
}
```

Output :

2 2.2000 C

```
void main()
{
      int Num  = 65;
      clrscr();
      printf("%c %d", Num, Num);
}
```

Output :

A 65

```
void main()
{
      int Num  = 7;
      clrscr();
      printf("%f", Num);

}
```

Output :

Error Message : "Floating points formats not linked"

Parul® University

scanf() function

- scanf() function reads all the types of data values.
- It is used for runtime assignment of variables.
- The scanf() statement also requires conversion symbol to identify the data to be read during the execution of the program.
- The scanf() stops functioning when some input entered does not match format string.

scanf() function

Syntax :

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

- Scanf statement requires '&' operator called address operator
- The address operator prints the memory location of the variable
- scanf() statement the role of '&' operator is to indicate the memory location of the variable, so that the value read would be placed at that location.

scanf() function

- The scanf() function statement also return values. The return value is exactly equal to the number of values correctly read.
- If the read value is convertible to the given format, conversion is made.

```
void main()
{
    int a;
    clrscr();
    printf("Enter value of 'a' : ");
    scanf("%d", &a);
   printf("A : %d",a);
}
```

OUTPUT

```
Enter value of 'a' : 8
A : 8
```

Data Type		Format string
Integer	Short Integer	%d or %i
	Short unsigned	%u
	Long signed	%ld
	Long unsigned	%lu
	Unsigned hexadecimal	%u
	Unsigned octal	%o
Real	Floating	%f or %g
	Double Floating	%lf
Character	Signed Character	%c
	Unsigned Character	%c
	String	%s
Octal number		%o
Displays Hexa decimal number in lowercase		%hx
Displays Hexa decimal number in lowercase		%p

Parul® University

Escape Sequence

- printf() and scanf() statement follows the combination of characters called escape sequence
- Escape sequence are special characters starting with '\'

Escape Sequence	Use	ASCII value
\n	New Line	10
\b	Backspace	8
\f	Form feed	12
\'	Single quote	39
\\	Backslash	92
\0	Null	0
\t	Horizontal Tab	9
\r	Carriage Return	13
\a	Alert	7
\"	Double Quote	34
\v	Variable tab	11
\?	Question mark	63

```
void main()
{
    int a = 1, b = a + 1, c = b + 1, d = c + 1;
    clrscr();
    printf("\t A = %d\nB = %d \'C = %d\'",a,b,c);

}
```

OUTPUT

A = 1

B = 2 'C = 3'

Unformatted Functions

- C has three types of I/O functions
  - Character I/O
  - String I/O
  - File I/O
  - Character I/O

getchar

- This function reads a character type data from standard input.
- It reads one character at a time till the user presses the enter key.
- Syntax

```
VariableName = getchar();
```

- Example

```
char c;
c = getchar();
```

putchar

- This function prints one character on the screen at a time, read by the standard input.
- Syntax
  - putchar(variableName)
- Example

```
char c = 'C';
putchar(c);
```

getch() and getche()

- These functions read any alphanumeric character from the standard input device.
- The character entered is not displayed by the getch() function.
- The character entered is displayed by the getche() function.
- Exampe
  - ch = getch();
  - ch = getche();

gets()

- This function is used for accepting any string through `stdin` keyword until enter key is pressed.
- The header file `stdio.h` is needed for implementing the above function.
- Syntax

```
char str[length of string in number];
gets(str);
```

```
void main()
{
        char ch[30];
        clrscr();
        printf("Enter the string : ");
          gets();
          printf("\n Entered string : %s", ch);
}
```

puts()

- This function prints the string or character array.
- It is opposite to gets()

```
char str[length of string in number];
gets(str);
puts(str);
```


Thank
You