Quadrant II – Transcript and Related Materials

Programme: F.Y.B.Sc

Subject: Computer Science

Paper Code: CSC101

Paper Title: Programming Fundamentals Using C

Unit:5-Overview Of C

Module Name: Basic I/O

Module No: 27

Name of the Presenter: Ms Prasanna Kothawale

Notes:

Basic Input Output Functions

- ***When a programmer says input, it would mean that they are feeding some data or info in the program. They can give this input from the command line or in the form of any file. C programming provides a set of built-in functions to read the given input and feed it to the program as per requirement.***

- ***On the other hand, when a programmer says output, they mean displaying some data and information on the printer, the screen, or any other file. C programming provides a set of built-in functions to output the data on the computer screen as well as to save it in text or binary files.***

There are mainly two types of these functions available in the C language:

1. ***Formatted:***

The formatted functions basically present or accept the available data (input) in a specific format. The standard library in C contains various

functions for the input-output operations. The scanf() and printf() out of these functions help a programmer format the functions in their desired format.

2.Unformatted:

unformatted functions are not capable of controlling the format that is involved in writing and reading the available data.eg getchar(),putchar(),gets(),puts()

- All the C language programs treat all of the devices in the program in the form of files.
- Thus, every display device is also addressed the same way the program addresses a file.
- In such a case, when a program starts executing, the following three files get opened up automatically for providing the program with access to the device's screen and computer.
- The file pointers are the means to access the file for reading and writing purpose. This section explains how to read values from the screen and how to print the result on the screen.

The format specifier

- We use the format specifiers when we are trying to print the values available with different data types with the use of printf() statement.
- we also use these format specifiers when taking the input from the users with the use of the scanf() function.
- It is because, here, we must let the program know what types of input it should be expecting from its user.

Format Specifier	Datatype
%d, %i	int
%c	char
%lf	double

%f	float
%hd	short int
%li	long int
%u	unsigned int
%lu	unsigned long int
%lli	long long int
%llu	unsigned long long int
%lf	long double
%c	signed char
%c	unsigned char
Symbol of Specifier	Meaning of Specifier
/t	Used by a program for tab spa total of 8 spaces)
/n	Used by a program for a new l linefeed return)

Output with printf()

- The printf() function,is part of the standard C library, is perhaps the most versatile way for a program to display data onscreen.
- The printf() function basicallly gets defined in the header file stdio.h, and we use it for showing the standard output (the output available on the console).
- Printing a text message onscreenis simple.
- Call the printf() function,passing the desired message enclosed in double quotation marks
- For example, to display an error that has occurred! onscreen, the user write the following:
- printf("An error that has occurred!");

- In addition to text messages, however, he frequently needs to display the value of program variables. This is a little more complicated than displaying only a message. It accepts a string parameter (called the format string), which specifies a method for rendering a number of other parameters (of which there typically may be arbitrarily many, of a variety of types) into a string.

printf()

For example, suppose the user's want to display the value of the numeric variable x onscreen, along with some identifying text.

Furthermore, he wants the information to start at the beginning of a new line. The printf() function as shown below:

```
printf("\nThe value of x is %d", x);
```

\n represents a new line character.

The resulting screen display, assuming that the value of x is 12, would display the following:

- The value of x is 12.
- In this example, two arguments are passed to printf().
- The first argument is enclosed in double quotation marks and is called

the format string.

- The second argument is the name of the variable (x) containing the value to be printed.
- we used the format specifier %d to print an integer

Taking Input with scanf()

Just as most programs need to output data to the screen, they also need to input data from the keyboard.

The most flexible way the program can read numeric data from the keyboard is by using the scanf() library function.

The scanf() function reads data from the keyboard according to a specified format and assigns the input data to one or more program variables.

- For example:
- The statement reads a decimal integer from the keyboard and assigns it to the integer variable x as shown below:

- scanf("%d", &x);
- The '%' indicates that the conversion specification follows: The 'd' represents the data type and indicates that the number should be read as a integer.
- The '&' is 'C' unary operator that gets the memory address of the variable following it. The user will read more about this operator and its associated operator '*', in the section 'Pointers' in the course.
- Likewise, the following statement reads a floating point value from the keyboard and assigns it to the variable rate: scanf("%f", &rate); The 'f' represents the data type and indicates that the number should be read as a float.