

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course: Computer Programming Lab
Course Code: CS201
B. Tech. I Sem. (CSE, ECE, MECH, CE, CHE)

Lab-6

Aim: Data types, Format specifiers and Operators

Data Types: These are the declarations for the user variables which determine the type and size of data associated with variables. Applying **qualifiers** with data types change their characteristics. There are **sign** (i.e. **signed** and **unsigned**) and **size** (i.e. **short** and **long**) qualifiers. **Exact size of data types depend on the compiler.** Normally, size qualifiers cannot be applied with **char** and **float** data types while sign qualifiers are not applied with **float**, **double** and **long double**.

Format Specifiers: These are used to specify the kind of data being passed as an argument to the input/output (printf and scanf) functions. The format specifier in c is a special character that starts with a **percentage sign (%)** and continues with a **letter** or **group of letters** that denotes the data type. Following table shows the different data types with qualifiers along with their format specifiers:

Data Type	Keyword	Qualifier	Declaration Syntax	Format Specifier	Size in Byte
Character	char		char	%c	1
		unsigned	unsigned char		
		signed	signed char		
Integer	int		int	%d, %i	2 or 4
		unsigned	unsigned int	%u	
		signed	signed int	%d, %i	
		short	short int	%hd	2
		unsigned short	unsigned short int	%hu	
		signed short	signed short int	%hd	
		long	long int	%ld, %li	4
		long long	long long int	%lld, %lli	8
		unsigned long	unsigned long int	%lu	4
		signed long	signed long int	%ld, %li	4
Real	float		float	%f	4
	double		double	%lf	8
		long	long double	%Lf	10
String				%s	

Note: (i) If an expression contains mixture of different data types then automatic conversion (known as **implicit type casting**) takes place from **lower to higher type hierarchy**. In C language, ascending order hierarchy is as: **int-> unsigned->long->unsigned long->double->long double**.

(ii) Forceful (**explicit type casting**) conversion is used to convert declared data type of a variable into other data type. Syntax: **(type) expression or variable**. For exp. **(float) i** converts into float type from integer i.

Operators: These are the types of symbols that inform compiler for performing some specific arithmetic, logical or any other types of functions by following predefined **precedence** and **associativity**. The precedence of operators determines the order in which operations are performed in an expression. **Operators with higher precedence are evaluated first.** Associativity determines the **direction** in which the expression will be evaluated. Following table shows category of operators used in the C-language:

Category	Operator	Operation	Precedence	Associativity
Highest precedence	() [] -> :: .	Function call Array subscript C indirect component selector C scope access/resolution C direct component selector	1	L → R (left to right)
Unary	! ~ + - ++ -- & * sizeof	Logical negation (NOT) Bitwise (1's) component Unary plus Unary minus Preincrement or postincrement Predecrement or postdecrement Address Indirection (return size of operand, in bytes)	2	R → L (right to left)
Member access	. ->	Dereference Dereference	3	L → R
Multiplication	* / %	Multiply Divide Remainder (modulus)	4	L → R
Additive	+ -	Binary plus Binary minus	5	L → R
Shift	<< >>	Shift left Shift right	6	L → R
Relational	< <= > >=	Less than Less than or equal to Greater than Greater than or equal to	7	L → R
Equality	== !=	Equal to Not equal to	8	L → R
Bitwise AND	&	Bitwise AND	9	L → R
Bitwise XOR	^	Bitwise XOR	10	L → R
Bitwise OR		Bitwise OR	11	L → R
Logical AND	&&	Logical AND	12	L → R
Logical OR		Logical OR	13	L → R
Conditional	?:	(a?x:y means "if a then x, else y")	14	R → L
Assignment	= *= /= %= += -= &= ^= = <<= >>=	Simple assignment Assign product Assign quotient Assign remainder (modulus) Assign sum Assign difference Assign bitwise AND Assign bitwise XOR Assign bitwise OR Assign left shift Assign right shift	15	R → L
Comma	,	Evaluate	16	L → R

Execute following program examples, see outputs and learn basic programming concepts.

Example#1:

*/*Conversion of temperature unit from Celsius to Fahrenheit*/*

```
#include<stdio.h>
int main ()
{
    float cls, fht;
    printf("Enter temperature in Celsius : ");
    scanf("%f", &cls);
    fht = (9.0/5.0)*cls+32;
    printf("Temperature in Fahrenheit is=%f\n", fht);           //Prints 6 digits after decimal point by default
    printf("Temperature in Fahrenheit is=%0.3f\n", fht);       //Prints 3 digits after decimal point
    printf("Temperature in Fahrenheit is=%d\n", (int) fht);    //Explicit type casting to print only integer part
    printf("Temperature in Fahrenheit is=%e\n", fht);          //Prints in exponential format
    return 0;
}
```

Example#2:

/ Working with Relational, logical and bitwise operators */*

```
#include <stdio.h>
int main()
{
    int a = 5, b = 5, c = 10, result;

    result = (a == b);                                     // Relational and Logical operators
    printf(" %d \n", result);
    result = (a == b) || (c < b);
    printf(" %d \n", result);
    result = !(a == b);
    printf("!(a == b) is %d \n", result);
    result = (a != b) || (c > b);
    printf("(a != b) || (c > b) is %d \n", result);

    result=(a&b)|c;                                       //Bitwise Operators
    printf("a&b|c is %d \n", result);
    result=(a|b)|(a&c);
    printf("(a|b)|(a & c) is %d \n", result);
    result=!c&(!a^b);
    printf("!c & (!a ^ b) is %d \n", result);
    return 0;
}
```

For first turn of the week:

Exercise#1: Write a C language program using variable values $a = 8$, $b = 29$, $c = 15.35$ and $d = 7.254$ to find:

- Sum and Average of all four numbers
- Sum and Average of integer part of all numbers
- Multiplication ($c * d$) upto two decimal point
- Quotient and Remainder of (b/a)
- Multiplication ($a * b$) in octal format using **%o** format specifier
- Multiplication ($a * b$) in hexadecimal format using **%x** or **%X** format specifier

Exercise#2: Write a C language **user input** program to evaluate following expressions. Verify results with value of variables $a=2$, $b=3$, and $c=5$.

- Arithmetic Expression: $2a \left[\log_{10} \frac{4a^2}{b} \right] + |b^2 - c^2| - 5\sqrt{c} + \cos 30^\circ$. **(Result=8.593813)**
Include additional header files **math.h** and **stdlib.h** to use in-built mathematical functions.
Syntax: **pow (a, b)**, **sqrt (a)**, **log10 (a)**, **cos (a)**, **abs (a)** etc.
- Logical Expression: $(a \bar{b})(a + c) + abc + \bar{b}c$ **(Result=1)**
- Repeat part (ii) using bitwise operators **(Result=7)**

Exercise#3: The distance between two cities is measured in kilometer. Write a C language **user input** program to print measured distance in meter, centimeter, feet, and inch. Draw the flowchart also in your practical book. (1 meter = 3.3 feet; 1 feet = 12 inch)

For second turn of the week:

Exercise#4: Suppose your computer monitor consumes 60 watt per hour, CPU consumes 80 watt per hour and all other hardware components consumes total of 40 watt per hour. Write a C language **user input** program to calculate **monthly consumed units** and **payable bill** @ Rs 8 per unit if computer system is used 12 hours/day. Draw the flowchart also in your practical book. (One unit = 1000 Wh)

Formula:
$$\text{Monthly Consumed Units} = \frac{30 \times \text{Total Power Consumed in Watt}}{1000}$$
$$\text{Monthly Payable bill} = \text{Monthly Consumed units} \times \text{Rate per unit} \times \text{Hours of use per day}$$

Exercise#5: The resolution of full HD computer monitor is 1920×1080 pixels. Write a C language **user input** program to calculate **total number of bits** (in Mb) required in one color image and find **how long** it will take to transfer 100 color images using network with data transfer rate of 512 Kb/second. Draw the flowchart also in your practical book. (One pixel of color image = 24 bits (8 bits per color of RGB)).

Formula:
$$\text{No. of bits in one frame (in Mb)} = \frac{\text{Pixels in a row} \times \text{Pixels in a column} \times \text{Bits per pixel}}{1024 \times 1024}$$
$$\text{Transfer time in second} = \frac{\text{No. of images} \times \text{No. of bits in one frame of the image}}{\text{Data transfer rate in bits per second}}$$

Exercise#6: A cashier has currency notes of denominations 10, 50 and 100. If a customer withdraws an amount in tens, hundreds or thousands. Write a C language **user input** program to find the total number of currency notes of each denomination the cashier will have to give to the withdrawer. Draw the flowchart also in your practical book.

Practice Questions

(No need to include in your Practical Book)

Q.1 Write a C program to print the sum of n natural numbers. The sum of first ' n ' natural numbers is given by:-

$$1 + 2 + 3 + 4 \dots n = \frac{n(n+1)}{2}$$

Q.2 Write a C program to print the sum of the square of n natural numbers. The sum of squares of first ' n ' natural no's-

$$1^2 + 2^2 + 3^2 + 4^2 \dots n^2 = \frac{n(n+1)(2n+1)}{6}$$

Q3. Write a C program to print the sum of the cube of n natural numbers. The sum of cubes of first ' n ' natural no's-

$$1^3 + 2^3 + 3^3 + 4^3 \dots n^3 = \frac{n^2(n+1)^2}{4}$$

Q.4 The length & breadth of a rectangle and radius of a circle are input through the keyboard. Write a C program to calculate the area & perimeter of the rectangle, and the area & circumference of the circle.

Q5. Write program that takes 3 digit number as an input. Program should compute and display the sum of digits in a number.

Q6. Write program that takes 3 digit number as an input. Write a C program to reverse the number and to obtain the sum of the first and last digit of this number.

Q7. Write a C program to find the gross salary of an employee. Ask user to enter the basic salary only. Assume Dearness Allowance (DA) = 40% of the Basic Salary, House Rent Allowance (HRA) = 80% of DA, and Bonus= 25% of HRA.

$$\text{Gross Salary} = \text{Basic Salary} + \text{DA} + \text{HRA} + \text{Bonus}$$

Q8. Two numbers are input through the keyboard into two locations X and Y. Write a C- program to interchange the contents of X and Y. (a) using third variable (b) without using third variable.,