Anil Neerukonda Institute of Technology & Sciences
Department of Computer Science & Engineering (AI & ML, DS)

CSE117 Problem Solving with C

Handout - Lab Session - 2

Objective :

- To write your first C program.
- To be able to create good identifiers for objects in a program.
- To be able to list, describe and use the C basic data types.
- To be able to use simple input and output statements.
- To be able to perform simple arithmetic operations.

Pre Lab : Go through the concepts of data types, variables, constants, input and output statements.

During Lab : Solve all the exercise problems. You should work on the additional set of programs only after completing this week's tasks.

Post Lab : Take the quiz.

The Basic Data Types

The most fundamental element of the C language is the Expression.
Expressions in C are substantially more flexible and powerful than in many other computer languages. Expressions are formed from the atomic elements.

★ Data
★ Operators

Data may be represented by variables, constants or values returned by functions.
C defines five fundamental data types.

	Data Type	keyword
1.	Character	char
2.	Integer	int
3.	Floating-point	float
4.	Double floating-point	double
5.	Value less	void

The size and the range of these data types may vary among processor types and compilers.
However, in all cases an object of type char is 1 byte.

Modifying the basic types
signed
unsigned
long
short

Type	Typical size in bytes	Range
char	1	-127 to 127
unsigned char	1	0 to 255
signed char	1	-127 to 127
short int	2	-32767 to 32767
unsigned short int	2	0 to 65, 535
signed short int	2	-32767 to 32767
int	2 or 4	-32767 to 32767
unsigned int	2 or 4	0 to 65, 535
signed int	2 or 4	-32767 to 32767
long int	4	-2,147,483,647 to 2,147,483,647
signed long int	4	-2,147,483,647 to 2,147,483,647
unsigned long int	4	0 to 4, 294, 967, 295
long long int	8	$-(2^{63} - 1)$ to $(2^{63}-1)$
unsigned long long int	8	$2^{64}-1$
float	4	1E-37 to 1E +37 (6 digits of precision)
double	8	Ten digits
long double	10	Ten digits

- By default integer declaration assumes a signed number. So, int and signed int are the same.
- The most important use of signed is to modify ‘char’ in implementations in which char is unsigned by default.

Constants

A constant is data whose value cannot be changed.

Constants can be coded in three different ways: as literals, as define commands, as memory constants.

Variables

Variables are named areas of memory used to hold data.

Variables must be declared and defined before being used in C.

Input/Output

To input data through the keyboard and to output data through the monitor, we use the standard formatted input/output functions.

scanf is a standard input function for inputting formatted data through the keyboard.

printf is a standard output function for outputting formatted data to the monitor.

To insert data into the stream, we use a conversion specification tha contains a start token (%), a conversion code, and upto four optional modifiers as shown below. Only the field- specification token(%) and the conversion code are mandatory.

%	Flag	Minimum Width	Precision	Size	Code

** Refer to your class notes for more information.

Exercise 1 : Use signof operator to verify the size of each of the above mentioned data types.

Exercise 2 : Execute the following printf statements one at a time and note down the outputs.

i. printf("%d%c%f ", 23,'z',4.1);
ii. printf("%d %c %f ", 23,'z',4.1);
(What is the difference between the outputs of the above two statements ?)
iii. int num1 = 23;
float num2 = 4.1;
char zee = 'z';
printf("%d %c %f ", num1, zee, num2);
iv. printf("%d\t %c\t %5.1f\n ", 23, 'z', 14.2);
printf("%d\t %c\t %5.1f\n ", 107, 'A', 53.6);
printf("%d\t %c\t %5.1f\n ", 1754, 'F', 122.0);
printf("%d\t %c\t %5.1f\n ", 3, 'P', 0.2);
In addition to the conversion specifications, note the tab character (\t) between the first and second, and second and third conversion specifications. Since the data are to be printed in separate lines, each format string ends with a new line (\n).

Exercise 3: Write a printf statement to print the statement,
"My favorite number is 25", using a conversion specification.

Exercise 4: Write a printf statement to print " The number is 23". Note that there are 5 blank spaces before the number 23. The first space comes from the space after 'is' and before the % in the format string. The other four comes from the width in the conversion specification.

Exercise 5: Take a number 233.12 and check what will be printed with %6.2f, %8.2f, %08.2f.

Exercise 6: Find the errors in the following printf statements

i. printf("%d %d %d \n", 44, 55);
ii. printf("%d %d \n", 44, 55, 66);
iii. float x = 123.45;
printf("The data are : %d\n", x);

Exercise 7: Find the errors in the following scanf statements

i. int a=0;
scanf("%d", a);
printf("%d\n", a);
Input : 234
Output : ?
ii. float a = 2.1;
scanf("%5.2f ", &a);
printf("%5.2f ", a);

Input : 74.35
Output : ?

iii.
```
int a;
int b;
scanf("%d%d%d", &a, &b);
printf("%d %d \n", a,b);
```
Input : 74 35 46
Output : ?

iv.
```
int a = 1;
int b = 2;
int c = 3;
scanf("%d%d", &a, &b, &c);
printf("%d %d\n", a, b, c);
```
Input : ?
Output : ?

Exercise 8: Write a C program to prompt the user to input 3 integer values and print these values in forward and reversed order, as shown below.

Input : Please enter three numbers: 23 57 64

Output: Numbers forward : 23
57
64
Numbers backward : 64
57
23

Exercise 9 : Write a C program to produce the output below.

x value	y value	expressions	results
20	10	x=y+4	x=14
20	10	x=y-3	x=7
20	10	x=y*4	x=40
20	10	x=x/5	x=2
20	10	x=x%y	x=0

Exercise 9: Your summer surveying job requires you to study some maps that give distances in kilometers and some that use miles. You and your coworkers prefer to deal in metric measurements. Write a program that performs the necessary conversion.

Input: miles /*the distance in miles*/
Output : kms /*the distance in kilometers*/

Relevant formula : 1 mile = 1.609 kilometers

Design, algorithm, flow chart, program using the above data requirements for the given problem. Try the sample test cases given below.

Sample Test Cases	Input	Output
Test case 1	10	16.09
Test case 2	2	3.218

Exercise 10: Supermarket coin processor

You are drafting software for the machines placed at the front of supermarkets to convert change to personalized credit slips. In this draft, the user will manually enter the number of each kind of coin in the collection, but in the final version, these counts will be provided by the code that interfaces with the counting devices in the machine.

Problem input

```
char first, middle, last  /*customer's initials*/
int dollars;              /*number of dollars*/
int quarters;             /*number of quarters*/
int dimes;                /*number of dimes*/
int nickles;              /*number of nickles*/
int pennies;              /*number of pennies*/
```

Problem outputs

```
int total_dollars;        /*total dollar value*/
int change;               /*leftover change*/
```

Additional program variables

```
int total_cents;          /*total value in cents*/
```

** Note: Pennies - 1 cent ; Nickels - 5 cents; Dimes - 10 cents; Quarters - 25 cents.; Dollar - 100 cents

Design, algorithm, flow chart, program using the above data requirements for the given problem. Try the sample test cases given below.

Testing tip : to test this program, try running it with a combination of coins that yield an exact dollar amount with no leftover change. For example, 1 dollar, 8 quarters, 0 dimes, 35 nickels, 25 pennies should yield a value of 5 dollars and 0 cents. Then increase and decrease the quantity of pennies by 1(26 and 24 pennies) to make sure that these cases are also handled properly.

Sample Test Cases	Input	Output
Test case 1	Your 3 initials : ABP Please enter your coin information. Number of dollars : 2 Number of quarters : 14 Number of dimes : 12 Number of nickels : 25 Number of pennies : 131	ABP coin credit Dollars : 9 Change : 26 cents
Test case 2	Your 3 initials : ABP Please enter your coin information. Number of dollars : 3 Number of quarters : 12 Number of dimes : 14 Number of nickels : 50 Number of pennies : 175	ABP coin credit Dollars : 11 Change : 65 cents

Exercise 11: Given the following pseudo code, write a program that executes it.

a. read x
b. read y
c. compute p=x*y
d. compute s=x+y
e. total=s2+p*(s-x)*(p+y)
f. print total