



JECRC Foundation



**JAIPUR ENGINEERING COLLEGE
AND RESEARCH CENTRE**

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

Year & Semester - B.Tech I year (I Semester)

Subject - Programming for Problem Solving

Presented by - Ms. Abhilasha /Ms. Yogita Punjabi/

Mr. Gajendra Sharma

Designation - Asst. Professor

Department - Computer Science (First Year)

VISSION OF INSTITUTE

To become a renowned centre of outcome based learning, and work towards academic, professional, cultural and social enrichment of the lives of individuals and communities.

MISSION OF INSTITUTE

- ❖ Focus on evaluation of learning outcomes and motivate students to inculcate research aptitude by project based learning.
- ❖ Identify, based on informed perception of Indian, regional and global needs, the areas of focus and provide platform to gain knowledge and solutions.
- ❖ Offer opportunities for interaction between academia and industry.
- ❖ Develop human potential to its fullest extent so that intellectually capable and imaginatively gifted leaders may emerge.

Programming for Problem Solving : Course Outcomes

Students will be able to:

CO1: Understand concept of low-level and high-level languages, primary and secondary memory. Represent algorithm through flowchart and pseudo code for problem solving.

CO2: Represent and convert numbers & alphabets in various notations.

CO3: Analyze and implement decision making statements and looping.

CO4: Apply pointers, memory allocation and data handling through files in 'C' Programming Language.

Pointer

Pointer is the variable which stores the address of the another variable

Declaration of pointer :

syntax : datatype *pointername;

Example :

int *ptr;

char *pt;

syntax :

pointer variable name=&variable name;

For Example :

int *p,quantity=20;

p=&quantity;

Variable	Address
Quantity	500
P	5048

Example

```
#include <stdio.h>
int main()
{
int var = 5;
printf("var: %d\n", var);
printf("address of var: %p", &var);
return 0;
}
```

Output

var: 5 address of var: 2686778

Pointer Syntax

Here is how we can declare pointers.

`int* p;` Here, we have declared a pointer `p` of `int` type.

You can also declare pointers in these ways.

`int *p1; int * p2;`

Assigning addresses to Pointers

Let's take an example.

`int* pc, c; c = 5;`

`pc = &c;`

Here, `5` is assigned to the `c` variable. And, the address of `c` is assigned to the `pc` pointer.

Example of Pointer demonstrating the use of & and *

```
#include <stdio.h>
void main()
{
int *p; int var = 10;
p= &var;
printf("Value of variable var is: %d", var);
printf("\nValue of variable var is: %d", *p);
printf("\nAddress of variable var is: %p", &var);
printf("\nAddress of variable var is: %p", p);
printf("\nAddress of pointer p is: %p", &p);
}
```

Output:

Value of variable var is: 10 Value of variable
var is: 10 Address of variable var is:
0x7fff5ed98c4c Address of variable var is:
0x7fff5ed98c4c Address of pointer p is:
0x7fff5ed98c50

Example2: Creating a void user defined function that doesn't return anything.

```
#include <stdio.h>
void introduction()
{
printf("Hi\n");
printf("My name is Abhilasha\n");
printf("How are you?");
}
int main()
{
introduction();
return 0;
}
```

Output:

Hi My name is Abhilasha How are you?

Advantage of pointer

- 1) Pointer **reduces the code** and **improves the performance**, it is used to retrieving strings, trees, etc. and used with arrays, structures, and functions.
- 2) We can **return multiple values from a function** using the pointer.
- 3) It makes you able to **access any memory location** in the computer's memory.

There are many applications of pointers in c language.

1) Dynamic memory allocation

In c language, we can dynamically allocate memory using malloc() and calloc() functions where the pointer is used.

2) Arrays, Functions, and Structures

Pointers in c language are widely used in arrays, functions, and structures. It reduces the code and improves the performance.

Pointer to function in C

As we discussed in the previous chapter, a pointer can point to a function in C. However, the declaration of the pointer variable must be the same as the function. Consider the following example to make a pointer pointing to the function.

```
#include<stdio.h>
int addition ();
int main ()
{
    int result;
    int (*ptr)();
    ptr = &addition;
    result = (*ptr)();
    printf("The sum is %d",result);
}
int addition()
{
    int a, b;
    printf("Enter two numbers?");
    scanf("%d %d",&a,&b);
    return a+b;
}
```

Output

Enter two numbers?10 15 The sum is 25

Pointer to Array of functions in C

The pointer to an array of functions is a pointer pointing to an array which contains the pointers to the functions. Consider the following example.

```
#include<stdio.h>
int show();
int showadd(int);
int (*arr[3])();
int (**ptr)[3]();
int main ()
{
    int result1;
    arr[0] = show;
    arr[1] = showadd;
    ptr = &arr;
    result1 = (**ptr)();
    printf("printing the value returned by show : %d",result1);
    ((*ptr+1))(result1);
}
```

Cont...

```
int show()
{
    int a = 65;
    return a++;
}
int showadd(int b)
{
    printf("\nAdding 90 to the value returned by show: %d",b+90);
}
```

Output

printing the value returned by show : **65** Adding **90** to the value returned by
show: **155**

Example to swap two numbers using pointers

```
#include <stdio.h>
int main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before Swapping\nx = %d\ny = %d\n", x, y);

    a = &x;
    b = &y;

    temp = *b;
    *b = *a;
    *a = temp;
    printf("After Swapping\nx = %d\ny = %d\n", x, y);

    return 0;
}
```

**Output: Enter the
value of x and y
Before Swapping 5 ,4
After Swapping 4 ,5**

Bibliography

Programming in ANSI C by E.Balagurusamy, McGrawHill

Let us C Yashavant Kanetkar, BPB Publication

<https://www.programiz.com/c-programming/c-functions>

https://www.tutorialspoint.com/cprogramming/c_functions.htm

<https://beginnersbook.com/2014/01/c-tutorial-for-beginners-with-examples/>



JECRC Foundation



**JAIPUR ENGINEERING COLLEGE
AND RESEARCH CENTRE**

*Thank
you!*