



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch: B4

Roll No.: 16010122828

Experiment No. 1

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Implementation of Abstract Data Type

Objective: Implementation of ADT without using any standard library function

Expected Outcome of Experiment:

CO	Outcome
CO 1	Explain the different data structures used in problem solving.

Books/ Journals/ Websites referred:



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Abstract:-

(Define ADT. Why are they important in data structures?)

Abstract Data Type (ADT) is a type (or class) for objects whose behaviour is defined by a set of value and a set of operations.

The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented. It does not specify how data will be organized in memory and what algorithms will be used for implementing the operations. It is called “abstract” because it gives an implementation-independent view. The process of providing only the essentials and hiding the details is known as abstraction.

Abstract Data Type for :

Arrays

[for chosen data type write value definition and operator definition)

Arrays have been used for dynamic memory allocation

ptr is an array pointer of size 3.

ptr[0] has been pointed to integer variable a, which has a value of 100

ptr[1] has been pointed to integer variable b, which has a value of 200

ptr[2] has been pointed to integer variable c, which has a value of 300

Then the pointer (ptr[0], ptr[1], ptr[2]) has been used to multiply all the three values by 25.

Strings

Strings have been used for dynamic memory allocation

str1[100] is an array of type char, used to store the first string

str2[100] is an array of type char, used to store the second string

str3[100] is an array of type char, used to store the concatenated string



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

1. Operation on string

Code:

```
#include <stdio.h>

int main()
{
    char str1[100];
    char str2[100];
    printf("Enter first string: ");
    scanf("%s", str1);
    printf("Enter second string: ");
    scanf("%s", str2);
    char str3[100];
    int i = 0, j = 0;
    printf("\nFirst string: %s", str1);
    printf("\nSecond string: %s", str2);
    while (str1[i] != '\0') {
        str3[j] = str1[i];
        i++;
        j++;
    }
    i = 0;
    while (str2[i] != '\0') {
        str3[j] = str2[i];
        i++;
        j++;
    }
    str3[j] = '\0';
    printf("\nConcatenated string: %s", str3);
    return 0;
}
```



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Output:

Output

```
/tmp/J7lJjvLKlz.o  
Enter first string: abc  
Enter second string: def  
First string: abc  
Second string: def  
Concatenated string: abcdef
```



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

2. Operation on array

Code:

```
#include <stdio.h>

int main()
{
    int a,b,c;

    int *ptr[3];

    ptr[0]= &a;
    ptr[1]= &b;
    ptr[2]= &c;

    a=10;
    b=20;
    c=30;

    printf("value of a: %d, b: %d, c: %d\n",*ptr[0],*ptr[1],*ptr[2]);

    *ptr[0] *=25;
    *ptr[1] *=25;
    *ptr[2] *=25;

    printf("After multiplying 25\nvalue of a: %d, b: %d, c: %d\n",*ptr[0],*ptr[1],*ptr[2]);

    return 0;
}
```



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Output:

```
Output
/tmp/J71JjvLK1z.o
value of a: 10, b: 20, c: 30
After multiplying 25
value of a: 250, b: 500, c: 750
```

Implementation Details:

1. Enlist all the Steps followed, and various options explored

1) For array

Steps followed:

- i. Three integer variables, a, b, and c are declared
- ii. An integer array pointer of size 3 is declared
- iii. The address of all the three integer variables is assigned to the pointer, ptr
- iv. Values are assigned to all the three variables
- v. The value of three variables is printed by the conventional method, that is, by directly printing the three variables
- vi. Then, the values of the variables are accessed and changed (multiplied by 25) using the pointer
- vii. The changed values are accessed and printed using the pointer

2) For string

Steps followed:

- i. Two arrays of type char are declared
- ii. Both the strings to be concatenated are taken as input from the user
- iii. A third array of type char is declared
- iv. Both the strings which were taken as input, are printed
- v. Two variables, i and j are initialised to 0, which are used as counter to iterate through the arrays
- vi. While each character in the first string is not null, the characters are copied in the third array of type char
- vii. While each character in the second string is not null, the characters are copied in the third array of type char, starting from the index after the last character of the copied string str1.
- viii. After copying both the arrays, the element of the next index is set to null



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

- ix. The third array str3, which contains the concatenated string is printed

2. Explain your program logic and methods used.

First program

In this program, three variables are declared, and the values stored in them are accessed using a pointer. The pointer can be used to access the values stored in the variables. The pointer does not store the values that are contained in the variables, but contains the address of the variables, which means any change in the value of the variables can be reflected when the values are accessed by the pointer and vice versa. The pointer is used to access the values stored by variables, whose address is contained in them.

If ptr is an array pointer, then the syntax is:

```
ptr[0] = &a; //storing address of variable a
```

```
printf ("%d", *ptr[0]) //accessing the value stored in a
```

This method is also known as abstraction because we are providing only the essentials and hiding other data.

Second program

In this program, two strings are taken by the user and stored in str1 and str2 respectively. Then a third variable of type char (array) is declared. Using iterative method, the contents of the first array is copied into the third array. Then the contents of the second array are copied into this third array just after the contents of the first array. Now the third array contains the concatenated string.

This string is printed to obtain the concatenated string.

3. Explain the Importance of the approach followed by you

The approach followed in the above program is data abstraction. The advantages of using data abstraction are:

- i. It reduces the complexity of viewing the things.
- ii. Avoids code duplication and increases reusability.
- iii. Helps to increase security of an application or program as only important details are provided to the user.



K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Conclusion: -

Thus, abstract data type is implemented without using in built functions.