

# Array

Q What is Array? Explain with example:

Ans Array is a memory variable capable to store more than one data. Normally any variable can store one data. When we store new data in any variable then last stored data will be deleted.

Example :

```
void main ()
```

{

int a = 5;

a = 10;

printf ("%.d", a);

}

Output :

v = 10

The output of above program is "10" which shows that "5" is deleted. There is a possibility to store more than one data in one variable and it is known as "Array".

Array is a memory variable to store more than one same type of data. It is created like any other variable is created but we use "[n]" to define its size.

Example :

```
int arr [20];
```

size of array

Date \_\_\_\_\_

Here "arr" is array to store twenty (20) values. It will store from "0" and goes up to "19".

Example: A Program to Enter marks of 10 students:

```
#include < stdio.h >
```

```
#include < conio.h >
```

```
void main()
```

```
{
```

```
int marks[10];
```

```
int i;
```

```
printf("Please! Enter marks of 10 students:-");
```

```
for (i=0; i<10; i++)
```

```
{
```

```
scanf("%d", &marks[i]);
```

```
for (i=0; i<10; i++)
```

```
{
```

```
printf("Marks of %d student is %d", i, marks[i]);
```

```
}
```

```
getch();
```

Output :-

Please! Enter marks of 10 students :-

72

Date / /

82

93

76

54

9

77

99

100

92

Marks of student 1 is 72.

Marks of student 2 is 82

Marks of student 3 is 92

Marks of student 10 is 92

The above program will store marks of 10 students in a single variable and print them.

\* Uses of Array: Array reduces the burden of creating many variables in any program. Suppose we have to store roll no. of 100 students then we don't need to create 100 variables rather we can create one variable with 100 pockets. Moreover, array provides many advantages like we can store data using loops which reduces the size of program. We can retrieve, search, add, sort and many more things makes "Array" powerfull.

## # Important Points

### 1. Declaration of An Array (Syntax):

datatype and arrayname [size];  
 ↑ subscript box

### 2. Example:

```
int num [10]; ← integer array
float salary [20]; ← float array
char name [5]; ← character array
```

### 3. Memory Calculation:

size-of datatype \* size-of array.

e.g.

int num [10];
 2 \* 10 = 20 byte

float salary [20];

4 \* 20 = 80 byte

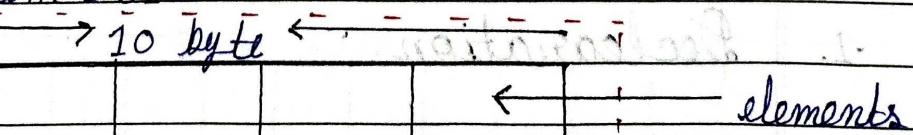
char name [5];

1 \* 5 = 5 byte, and so-on.

### 4. Memory Representation (RAM):

int num [5];

→ 10 byte



Date \_\_\_\_\_

## 5. Assignment of Array

int num[5];

num[0] = 10;

num[1] = 21;

num[2] = 72;

num[3] = 25;

num[4] = 32;

num[8] = 79; ~~X~~ // it assign value but it is disadvantage of array

## 5. Accessing of Array:

int x;

x = num[2];

now, x = 72;

int z;

z = num[8];

then printf("%d", z);

z = 692388 is a random value.