Dynamically allocate memory in C++

A new operator is used to dynamically allocate memory on the heap. Memory is allocated by new must be de-allocated using the delete operator.

Syntax of new is:
p_var = new type;
where p_var is a previously declared pointer of type typename. typename can be of any basic data type.

the allocated memory can be released using the following statement,

delete p_var;

new can also create an array:
p_var = new type [size];
In this case, size specifies the length of the one-dimensional array that is created.

the allocated memory can be released using the following statement,

delete[] p_var;

Example 1:

```
int  *p;

p=new int;
```

It allocates memory space for an integer variable. Also, the allocated memory can be released using the following statement,

```
delete p;
```

Example 2:

```
int *ptr;

ptr = new int[100];
```

It creates a memory space for an array of 100 integers. a[0] will refer to the first element, a[1] to the second element, and so on. In addition, in order to release the memory the following statement can be used,

```
delete[] ptr;
```

Example 3:

```
int **matrix , i, rows,columns;
```

…….

matrix = new int* [rows];

for (i=0; i<rows; i++)

matrix[i]= new int [columns];

It allocates memory space for a double dynamic array. Also, the release of memory occurs in the following manner:

for (i=0; i<rows; i++)

delete[] matrix[i];

delete[] matrix;