Dynamic allocation of Memory

By Avani M. Sakhapara
Assistant Professor,
IT Dept, KJSCE

Introduction

- C uses malloc() and calloc() functions to allocate memory dynamically and free() function to free dynamically allocated memory.
- In C++ **new** operator is used to **allocate the memory** dynamically and **delete** operator is used to **free** dynamically allocated memory

New Operator

- The new operator can be used to create objects of any type
- syntax:

pointer_variable = **new** data_type;

- pointer_variable is of type data_type
- Eg: int *p;

```
        p = new int;
                or
int *p = new int;
```

Initializing the memory

- *p = 25;
- The above statement assigns the value 25 to the newly created int object.
- Alternate method:

pointer_variable = **new** data_type(value);

- Eg: int *p = new int(25);

New operator to create 1D Array

- New can be used to create a memory space for any data type including user-defined types like arrays, structures and classes.
- Syntax to allocate memory for 1D Array:

```
pointer-variable = new data-type[size];
```

- Here size specifies the number of elements in an array.

New operator to create 1D Array

- Eg: int n=4;

  ```
  int  *p = new int [n];
  ```

- The above statement creates a memory space for an array of 4 integers.
- p[0] refers to the first element, p[1] refers to the second element and so on….

Delete Operator

- It is used to destroy the object created by new operator and release the memory.
- Syntax: delete pointer-variable
- Eg: delete p;
- To free dynamically allocated array, syntax is

delete []pointer-variable;

- Eg: delete []p

Advantages of new operator over malloc()

- Discussed in the lab