Dynamic Memory Allocation

- Uses `new` operator
  - `new <type>`
  - `new <type> (<init values>)`
  - `new <type>[size]`
  - `new` allocates space and returns a pointer to `<type>`

- Example
  
  ```
  int * p, q, r;
  p = new int;
  q = new int (2.718282);
  r = new int[100];
  ```
• **new** throws an exception or returns NULL if there is not enough space, so check
  
  ```
  p = new int[100000];
  if ( p )
    { count << "error -- out of memory" << endl; }
  ```

• **If you allocate, you must deallocate.**
  
  - Use the **delete** operator
  - `delete p;`
  - `delete[] p;`

• **Returning pointers from functions**
  
  - OK if data was passed as a param
  - OK if data was dynamically allocated inside function
  - NOT OK if data is declared locally