C++ Arrays

- An array is a collection of elements of same type.
- Elements of the array are placed in contiguous memory locations.
- Any random element of array can be accessed using its index.

Statically declared arrays

- Size of array is declared at compile time.
- Size of the array is fixed and cannot change.
- C++ compiler automatically handles the allocation and deallocation of memory for statically declared arrays.

Dynamically declared arrays

- Size of array can be declared at run time.
- Size of the array can be altered.
- Memory allocation and deallocation needs to be managed by programmer.

Syntax for statically declared arrays in C++

<data type> <variable name> [<size>];	Creates array of specified size. Elements of array contain garbage values.
<data type> <variable name> [<size>] {};	Creates array of specified size. Elements of array are initialized to default value of zero in case of array with numeric data type. In case of array of objects, default constructor is invoked by the C++ compiler.
<data type> <variable name> [<size>] = {};	Same as above. = is extra and can be omitted.
<data type> <variable name> [ ] { obj1, obj2, obj3, ... , objn };	Creates an array with specified values. The size of the array is deduced by C++ compiler based on number of specified values.
<data type> <variable name> [ ] = { obj1, obj2, obj3, ... , objn };	Same as above. = is extra and can be omitted.
<data type> <variable name> [<size>] { obj1, obj2, obj3, ... ,objn };	Creates an array with specified values. If number of specified values is greater than the specified size, then, it results in compilation error. If number of specified values is less than the specified size, then, the last remaining elements of the array are initialized with default value of zero in case of numeric data type; in case of objects, default constructor is automatically invoked by C++ compiler for the remaining last elements of the array.
<data type> <variable name> [<size>] = { obj1,obj2,obj3,...,objn };	Same as above. = is extra and can be omitted.