Programming for Engineers II
Class 42
STL: `vector`

Attique Dawood

April 29, 2014

Last Modified: Sunday 4th May, 2014, 18:09

1 Announcements

- None.

2 Vector Usage

- Header:

```
#include <vector>
using std::vector;
```

- Create vector object:

```
vector<int> intvector;
vector<myclassname> myclassvector;
```

- Initialization:

```
intvector.resize (20, 0); // int vector of size 20, all values intialized to 0.
```

- Get size: Total number of elements in vector.

```
int mysize = intvector.size();
cout << intvector.size();
```

- Insertion:

```
Single insertion at back: intvector.push_back (3); // Creates a new integer, NOT th
Single or multiple insertions at a specific location:
- intvector.insert (intvector.begin()+1, 1, 33); // Insert 33 at 2nd location.
- intvector.insert (intvector.begin(), 3, 1); // Insert 3 1's at the beginning.
```

- Access:

```
Operator [] can be used to access entries:
- int x = intvector[2]; // NOTE: This is NOT the same as insertion. Size remains sa
- intvector[5] = 22;
```

- Deletion:

```
Delete a single entry:
- myvector.erase (myvector.begin()+i); // Delete ith entry. Size is reduced by one.
Delete a range of entries:
- myvector.erase (myvector.begin()+1, myvector.begin()+5); // Delete elements in th
Delete vector:
- myvector.clear ();
```

3 Tasks

1. Create a dynamic array of integers using vector. Display contents after each step.

- Insert 10 random numbers.
- Display the contents.
- Delete first element.
- Delete 3rd , 4th and 5th elements.
- Insert 2 4's at beginning (index 0).
- Insert 4 -1's starting at second last location.
- Display final size.
- Delete vector.
- Resize the vector to 20 size initialized to -1's.
- Set values randomly. Size remains 20.

2. Create dynamic 2D and 3D arrays using vectors.

```
int rows = 3;
int cols = 5;
- vector<vector<int>> my2DArray (rows, vector<int>(cols)); // Declaration.
- Elements can be accessed using double subscript notation [][] in nested for loops.
```

3.1 Helpful Resources

- If string input is being skipped, use the following before getline() statement.

```
if (cin.peek() == '\n')
    cin.ignore();
getline (cin, stringname, '\n');
```

- getline() can be used to read from a txt file line by line.

```
fstream file("abc.txt", ios::in);
while (true)
{
    getline(file, tempstring, '\n');
    if (file.eof())
        break;
    cout << "Read: " << tempstring << endl;
}
file.close();
```

- To access the char* in string use stringname.c_str();

```
fstream file("abc.txt", ios::out);
string message = "this is a mesage. Writing this to file.";
file << message.c_str() << endl;
file.close();
```