MA122 - Computer Programming and Applications

Indian Institute of Space Science and Technology

April 21, 2017

Lecture 32

example


```
1  #include <iostream>
2  // function template prototype
3  template <typename T>
4  void Swap(T &a, T &b);
5  int main()
6  {
7  using namespace std;
8    int i = 10;
9    int j = 20;
10   cout << "i, j = " << i << ", " << j << ".\n";
11   cout << "Using compiler-generated int swapper:\n";
12
13   Swap(i,j); // generates void Swap(int &, int &)
14   cout << "Now i, j = " << i << ", " << j << ".\n";
15
16   double x = 24.5;
17   double y = 81.7;
```

example


```
1    cout << "x, y = " << x << ", " << y <<".\n";
2    cout << "Using compiler-generated double swapper:";
3    Swap(x,y); //generates void Swap(double &, double &)
4    cout << "Now x, y = " << x << ", " << y << ".\n";
5
6    return 0;
7  }
8
9  template <typename T>
10 void Swap(T &a, T &b)
11 {
12   T temp;
13   temp = a;
14   a = b;
15   b = temp;
16 }
```

fixed array


```
#include<iostream>
#include <vector>
int main()
{

int size = 10;
  int sarray[10];
  int *darray = new int[size];
  // do something with them:
  for(int i=0; i<10; ++i){
    sarray[i] = i;
    darray[i] = i;
  }
  // don't forget to delete darray when you're done
  delete [] darray;return 0;
}
```

Lecture 32

Vector template class


```
1  //How to use: include the appropriate header
2
3  #include<vector>
4  using std::vector;
5
6  //Specify the type by putting it between a pair of
       angle
7  //brackets following the template name
8
9  vector<T> v1; //vector that holds objects of type T;
10        //Default constructor v1 is empty
11
12 vector<int> ivec //ivec holds objects of type int
13
14 vector<Sales_item> Sales_vec //holds Sales_items
```

Vector template class


```
1  vector<T> v2(v1); // v2 is a copy of v1
2
3
4  vector<T> v3(n,i) //v3 has n elements with value i
5
6  vector<T> v4(n) //v4 has n copies of a value-
       initialized object
7
8  //vector is not a type; it is a template that we can
       use
9  //to define any number of types. Each vector type
       specifies an element
10 //type. Hence vector<int> and vector<string> are types
       .
```

Vector template class


```
1 vector<int> ivec1;
2 vector<int> ivec2(ivec1);
3 vector<string> svec(ivec1); //error: svec holds
      strings, not ints
```

Vector template class


```
1  #include<iostream>
2  #include <vector>
3  int main()
4  {
5
6  int size = 10;
7  std::vector<int> array(size); // make room for 10
       integers,
8  // and initialize them to 0
9  // do something with them:
10 for(int i=0; i<size; ++i){
11   array[i] = i;
12 }
13 return 0;
14 }
```

push_back


```
1  #include<iostream>
2  #include <vector>
3  int main()
4  {
5      std::vector<char> array;
6      char c = 0;
7      while(c != 'x'){
8      std::cin>>c;
9      array.push_back(c);
10     }
11
12 return 0;
13 }
```

reserve


```
1  #include<iostream>
2  #include <vector>
3  int main()
4  {
5      std::vector<int> array;
6      array.reserve(10); // make room for 10 elements
7      int c = 0;
8      while(c != -1){
9      std::cin>>c;
10     array.push_back(c);
11     }
12 return 0;
13 }
```

capacity


```
1  #include<iostream>
2  #include <vector>
3  int main()
4  {
5      std::vector<int> array;
6      int i = 999; // some integer value
7      array.reserve(10); // make room for 10 elements
8      array.push_back(i);
9      std::cout<<array.capacity()<<std::endl;
10     std::cout<<array.size()<<std::endl;
11 return 0;
12 }
```

Lecture 32

break example with do while


```
1  #include<iostream>
2  #include <vector>
3  int main()
4  {
5      std::vector<int> array;
6      array.reserve(15);
7      int a = 0;
8      do{
9          array.push_back(a);
10         std::cout << "value of a: " << array[a] << std
               ::endl;
11         a = a + 1;
12         if( a > 15) {
13             // terminate the loop
14             break;
15                     }
16     }while( a < 20 );
17 return 0;
```

Lecture 32

continue example with do while


```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      int a = 10;
6
7      do {
8          if( a == 15) {
9              // skip the iteration.
10             a = a + 1;
11             continue;
12         }
13         cout << "value of a: " << a << endl;
14         a = a + 1;
15     }while( a < 20 );
16
17     return 0;
18 }
```