How To Use the "Integer" Class?

```
// This sample program explains how to use the Integer class.
// A class in C++ defines a new type
// Associated with each class:
//     Data Members + Member functions( operations on objects of that class)
// For instance: The class Integer has an array of IntRep( Internal
// integer representation) to hold integer values.
// The operations on objects ( Variables ) of a class are the functions
// that manipulate the data members.
// There are two special operations: Constructors and Destructors
// Constuctors are used to initialize new objects with some initial valuses
// and Destructors are used to deinitialize these objects when they go out
// of scope.
// The operations that can be used on objects of type Integer are similar
// to those operations that are used with the built in data type (int)
// The rest of this file contains a sample declarations and manipulations
// of some Integer objects (variables):
// Path for the include files is: /usr/local/lib/g++-include

#include <iostream.h>     // C++ Standard Input/Output header file
#include <Integer.h>      // Declaration of Integer class


int  main(){

// Declaration  a variable (x) of type Integer
Integer x(atoIntRep("13874237894238948923783782389484723456789012345678901234567890123456
78901234567890123456789 0"));
```

```
// Declaration  a variable (z) of type Integer
Integer z;            // no initialization of object z

// We can do all kind of operations on integers
 z = x*y;             // Multiply x and y and store in z
 x += y ;             // x = x + y
 x++;                 // x = x + 1
 a.abs();             // absolute value of a
 // notice the C++ way of calling member functions (use the selection
 // operator .) to invoke any function

 Integer q = gcd(x,y);   // Check the listing of Integer.h for a complete
 // prototypes of all avaiable functions

 cout<< " Greatest common divisor of " << x << "and " << y << "is " << q <<endl;

 Integer c(13L);
 z = pow(x,c) % y;  // raise x to c mod y

 cout << "x =" << x << endl;  // cout is ostream object (used like  stdout in C)
 cout << "y =" << y << endl;
 cout << "z =" << z << endl;


 // Finally, to complie:
 // g++ your_file_name.C -o output_file
 // Notice that the g++ compiler expects filename.C extension (capital C)
 // inorder to invoke the C++ compiler.


return 0;
// Sample Output:
//Greates common divisor of 112639670004115492134771436710583512480638250835917
//94416466023961411111011111111101
// and 98765432109876543210987654321098789023849238490239042930092349490654321O
// 9876543210is 1
// x =11263967000411549213477143671058351248063825083591794416466023961411
//111011111111101
// y =9876543210987654321098765432109878902384923849023904293009234949065432109
//876543210
// z =1011604599536253895360966019982678556616319467010620551370822807196383165
```

```
//10278501

}
```