CS 101 Computer Programming and Utilization

Lecture 22

Operator Overloading

April , Tuesday 5 2011

Prof. R K Joshi
Computer Science and Engineering
IIT Bombay
Email: rkj@cse.iitb.ac.in

Revision

- uses of random numbers
- random numbers
  - pseudo random numbers
    - sequence repeats
  - random seed
- monte carlo method
- shell variables
- shell commands
- piping of shell commands
- redirection

Operator overloading

- define our own operators
- useful for developing operations on our classes
- e.g.we can define our own operations “<<” and “>>” on a class?
- class Array { .. } : A rich class implementing a dynamically growing array
- define << operation on this class?

Example

```
class Array {
 ...
 public:
     void  &operator << ( int x) {
                 // insert x at the end of the array
     }
};
```

A limitation

```
class Array {
 ...
 public:
     void  &operator << ( int x) {
                 // insert x at the end of the array
     }
};
Array a;
      a << x << y << z ;      // this will not work?
```

How does the second << invocation work in the nested invocations?

```
class Array {
 ...
 public:
     void  &operator << ( int x) {
                 // insert x at the end of the array
     }
};
Array a;
     ( (a  << x)<< y  ) << z ;
```

so what should be the return result of the invocation?

Object returns itself so that nested operations are possible

```
class Array {
 ...
 public:
     void  &operator << ( int x) {
                 // insert x at the end of the array
          …
          return *this;
     }
};
Array a;
     a << x << y   << z ;   // possible now      … over to demos
```