# CSE 4510/5241 HW3
## Submit server: course=dc, project=hw3
## Due 5pm, Mar 18, 2009

The objective is to design and implement (un)marshalling objects that have pointers using an IDL. The properties are:

1. An object that is pointed by multiple objects should NOT be marshalled more than once.

2. The number of objects and the "pointer structure" among objects remains the SAME on both hosts.

The IDL complier compiles the IDL file into the class definition and stub files. The library has class-definition-independent methods and additional data structures for (un)marshalling. The class defintion, stub, and library are compiled and linked with the user source. Your implementation marshalls, transmits, and unmarshalls a linked list.

1. each node has: name, age, city

2. data: www.cs.fit.edu/~pkc/classes/dc/hw3data.txt

Your design should be flexible to handle different field names, field orderings, numbers of fields (only int and string "basic" types), and different values in the objects.

```
--- Node.idl (Interface Definition Language file) ---
class Node
String name
int    age
String city
Node   next

--- Node.java/h: generated by the IDL compiler from Node.idl ---
public class Node
{
  String  name;
  int     age;
  String  city;
  Node    next;
};

--- NodeStub.java/c: generated by the IDL compiler from Node.idl ---
void marshallNode(Node node, byte[] msg) {...}
Node unmarshallNode(byte[] msg) {...}

--- Library.java/c ---
void marshallString(String str, byte[] msg) {... print marshalling str @ loc in msg and new/existing?}
void marshallInt(int num, byte[] msg) {...}
String unmarshallString(byte[] msg) {... print unmarshalling objID}
int unmarshallInt(byte[] msg) {...}
void printMarhsallObjectTable() {...}

--- Sender.java/c ---
sendList()
{
  Node list = buildList()  // from hw3data.txt
  print list
  marshallNode(list, msg)
  printMarshallObjectTable()
  print length of msg
  send(msg)
  receive(ack)
}

--- Receiver.java/c ---
receiveList()
{
  receive(msg)
  print length of msg
  send(ack)
  Node list = unmarshallNode(msg)
  printList()
}
```

**CSE 5241 students only**
Additional features for the same program:

- binary search tree (BST):

  1. bstNode.idl with class bstNode
  2. sendBst() in sender.java/c and receiveTree() in receiver.java/c
  3. buildBst() with hw3data.txt
  4. printBst): pre-order traversal, indent more at each level, one node per line

- String objects can be shared between the list and the tree.

What to turn in:

1. Detailed description of your "external marshalled format" (in the README file)
2. Compilation instructions (preferably makefile)
3. Source code
4. Sample session (`script` on unix)