Homework CS116 due Friday 02/02/01

1 Various questions (20pts)

Write four lines in english for each question.

1. How can you control the access to certain data fields ?

2. How can you specify to the compiler that an existing type can be implicitly converted to a type you define yourself ?

3. What happens when a type you define yourself is passed by value to a function ?

4. Why is it sometime usefull to use parameters passed by references instead of passing them by value ?

5. What would be the data fields for a `Node` class used for a list containing pairs of doubles ?

2 A polynomial class (80pts)

This class uses **dynamically allocated arrays for the coefficients**. Write all the member functions given below and be careful with memory management. The coefficient of the highest degree must be **always** different than 0, so that you **never** store useless coefficients in the representation of the polynomial. By convention the null polynomial will have a degree equal to -1. Note : the member functions are roughly sorted by difficulty.

```
class Poly {
  double *coeff;
  int degre;
public:
  // default constructor
  Poly();
  // built the polynomial from the degre and a list of coefficients
  Poly(int d, double *c);
  // copy constructor
  Poly(const Poly &p);
  // construct ONE polynomial equal to c*X^k
  Poly(double c, int k);
  // To convert a double to a polynomial of degre 0
  Poly(double x);
  // Destructor
  ~Poly();

  Poly &operator = (const Poly &p);
  bool operator == (const Poly &p) const;

  void print();
  Poly derivative() const;
  Poly operator * (const Poly &p) const;
  Poly operator + (const Poly &p) const;
};
```

So that we can execute (for example) the following `main()` :

```
int main() {
  // We initialize P to 5*X^3 + 1
  double x[] = {1, 0, 0, 5};
  Poly p(3, x);
  p.print();

  Poly q = p.derivative();
  p.print();

  // We use here the *, the + and the implicit conversion from double
  Poly r = p * q + (p + 2.0);
  r.print();
}
```