You must substitute the question marks for the appropriate class names. See if you can adapt the code to provide an implementation that can handle multiple memento’s. This functionality should be implemented within the originator.

```cpp
// First class
class ?;

class ?{
    friend class ?;

    private:
    {(){}
        int coordX;
        int coordY;
        int coordZ;
    }

    // Second class
    class ?{
    public:
    {() {
        token = 0;
    }

    void rememberMe(? m){
        /*We assume 'token' is not available/valid from outside*/
        if (token)
            delete token;
        token = m;
    }

    ? recallMe()
    {
        return token;
    }

    private:
    {* token;
    }

    // Third class
    class ?{
    public:
    {?(int x, int y, int z, string n) {
        x = x;
        y = y;
        z = z;
        name = n;
    }
```
void move(int x_, int y_, int z_)
{
    cout << "I am " << name << " and I am moving to: [" << x_ << ", " << y_ << ", " << z_ << "]" << endl;
    x = x_;
    y = y_;
    z = z_;;
}

void whereAreYou()
{
    cout << "My name is " << name << " and I am at: [" << x << ", " << y << ", " << z << "]" << endl;
}

?* checkPoint()
{
    cout << "returning Memento[" << x << ", " << y << ", " << z << "]" << endl;
    ?* memento = new ?();
    memento->coordX = x;
    memento->coordY = y;
    memento->coordZ = z;

    return memento;
}

void rollBack(?* memento)
{
    // This version assumes that a uses up
    // a memento and therefore has the
    // responsibility to deallocate the memory.
    cout << endl << "+++++++++++++++" << endl << "My name is at [" << x << ", " << y << ", " << z << "] before the reset." << endl;

    this->x = memento->coordX;
    this->y = memento->coordY;
    this->z = memento->coordZ;

    delete memento;

    cout << "?* name=" << " is at [" << x << ", " << y << ", " << z << "]" << " after the reset." << endl << "+++++++++++++++" << endl;
}

private:
    // The coordinates
    int x;
    int y;
    int z;

    // The name of the?
    string name;