IsingExperiment
Runs the experiment, sets IC, runs for a number of MC steps, Tell observers to take measurements

```
static void phase_diagram(int L);
static void hysteresis(int L);
static void movie(int L);
```

MetropolisHasting
Runs a Metropolis Hastings Algorithm on a model.

```
double _T;
std::mt19937 _gen;
std::uniform_real_distribution<> _distr;
```

```
MetropolisHasting(double T);
void run (int MCSteps,IsingModel* model);
void set_T(double T);
bool accept(double deltaE);
void set_seed(double seed);
```

Ising Model
Manages Spins, Calculates Energy.

```
std::vector<IsingSpin> _spins;

double _J;
double _H;
int _L;

std::mt19937 _gen;
std::uniform_int_distribution<> distr;
```

```
IsingModel(int L,double H,double J);

IsingSpin& get_random_spin();
IsingSpin& get_spin(int i);

double get_energy(IsingSpin& spin);


void set_seed(double seed)
void set_H(double H)
int get_system_size()
```

IsingSpin
Points up or down, can flip, knows its index.

```
bool _state = false;
int _index;
```

```
int get_state();
int get_index();

void flip();
void set_index(int i);
```

IsingObserver
Takes Measurments of an Ising Model

```
std::fstream outFile;
int _numMeasurements = 100;
```

```
IsingObserver(std::string fileName);
~IsingObserver();
virtual void takeMeasurement(IsingModel *subject,MetropolisHasting* MC, double value);
```

MagnetismIsingObserver
Measures Magnetism

```
virtual void takeMeasurement(...);
```

```
double m = 0;
for (int j = 0; j<_numMeasurements; j++){
    for (int i = 0; i<subject->get_system_size(); i++){
        m+=subject->get_spin(i).get_state();
    }
    MCAlgorithm->run(1,subject);
}
m=m/subject->get_system_size()/_numMeasurements;
outFile«value«","«m«std::endl;
```

MovieIsingObserver
Measures the entire state of the model

```
virtual void takeMeasurement(...);
```

```
outFile«value«',';

for (int i = 0; i<subject->get_system_size(); i++){
    outFile«subject->get_spin(i).get_state()«',';
}

outFile«std::endl;
```