

[Help](#)

```
/* Nothing is printed to files if the number of samples
 * exceeds 100000. Max File size is exceeded otherwise. */
#include <iostream>
#include <fstream>
#include <cmath>
#include <cstdlib>

#include "
href../../../../common/math/mcam/src/Option_h_src.pdfOption.hpp"
#include "
href../../../../common/math/mcam/src/performance_h_src.pdfperformance.hpp"
#include "
href../../../../common/math/mcam/src/basket_h_src.pdfbasket.hpp"
#include "
href../../../../common/math/mcam/src/asian_h_src.pdfasian.hpp"
#include "
href../../../../common/math/mcam/src/PremiaOption_h_src.pdfPremiaOption.hpp"

using namespace std;

mcam::Option::Option() { }
mcam::Option::~~Option() { }

/**
 * Creates the correct option according to name
 */
mcam::Option *mcam::instantiate_option(const Param &P)
{
    string Optiontype;
    mcam::Option *opt = NULL;
    P.extract("option type", Optiontype);

    if (Optiontype == "exchange" || Optiontype == "basket")
        opt = new mcam::BasketOption (P);
    else if (Optiontype == "geometric_put")
        opt = new mcam::GeometricPutOption (P);
    else if (Optiontype == "geometric_call")
        opt = new mcam::GeometricCallOption (P);
    else if (Optiontype == "bestof")
```

```

        opt = new mcam::BestOfOption (P);
    else if (Optiontype == "worstof")
        opt = new mcam::WorstOfOption (P);
    else if (Optiontype == "asian")
        opt = new mcam::AsianOption (P);
    else if (Optiontype == "moving_average")
        opt = new mcam::MovingAverageOption (P);
    else if (Optiontype == "performance")
        opt = new mcam::PerformanceOption (P);
    else if (Optiontype == "PremiaOption")
        opt = new mcam::PremiaOption (P);
    else
        printf("Option type not found.\ n");
    return opt;
}

```

```

mcam::Option::Option(const Param &P)
{
    P.extract("model size", size);
    P.extract("maturity", T);
    int dates = 1;
    int subticks = 1;
    P.extract("dates", dates);
    P.extract("sub ticks", subticks);
    subdates = subticks * dates;
}

```

```

void mcam::Option::print() const
{
    cout << " option size : " << size << endl;
    cout << " maturity time : " << T << endl;
}

```