

[Help](#)

```
#ifndef _BASKET_H
#define _BASKET_H

#include "pnl/pnl_matrix.h"
#include "math/ImportanceSampling_jl/src/parser.hpp"
#include "math/ImportanceSampling_jl/src/Option.hpp"

class BasketOption : public BaseOption
{
public:
    BasketOption();
    BasketOption(const Param &ParamTab);
    ~BasketOption();
    void print() const;

    double K;
    PnlVect *lambda;

    double payoff(const PnlMat *path_val);
};

class PerformanceOption : public BaseOption
{
public:
    PerformanceOption();
    PerformanceOption(const Param &ParamTab);
    ~PerformanceOption();
    void print() const;

    double K;
    PnlVect *lambda;

    double payoff(const PnlMat *path_val);
};

class BestOfOption : public BaseOption
{
public:
    BestOfOption();
```

```
BestOfOption(const Param &ParamTab);
~BestOfOption();
void print() const;

double K;
PnlVect *lambda;

double payoff(const PnlMat *path_val);
};

class WorstOfOption : public BaseOption
{
public:
    WorstOfOption();
    WorstOfOption(const Param &ParamTab);
    ~WorstOfOption();
    void print() const;

    double K;
    PnlVect *lambda;

    double payoff(const PnlMat *path_val);
};

class GeometricBasketOption : public BaseOption
{
public:
    GeometricBasketOption();
    GeometricBasketOption(const Param &ParamTab, bool is_call = true);
    ~GeometricBasketOption();
    void print() const;

    double K_;
    bool is_call_;
    double payoff(const PnlMat *path_val);
};
#endif
```