

UML

Project 3

① CityWeight

private:

cityWeight : double

public:

CityWeight()

CityWeight(double cityWeight)

getCityWeight() : double

setCityWeight(double cityWeight) : void

① Brute Force

private:

optimal : double

cw [20][20] : CityWeight*

elite : std::queue<int>

public:

BruteForce(CityWeight* cw [20][20])

factorial(int num) : int

permutation(int n, int arr[], int size) : void

optimalCheck(int arr[], int size) : void

getMostOptimalTime() : double

getMostOptimalRoute() : std::queue<int>

① File Reader

private:

public:

objectMaker(std::string fileName, CityWeight* cw [20][20]) : void

① Mutation Remastered

private:

int rows;

int cols;

int tours;

int generations;

int size;

double percentageOfMutations;

double optimalOne;

double optimalTwo;

std::queue<int> eliteOne;

std::queue<int> eliteTwo;

CityWeight* cw [20][20]

public:

MutationRemastered(int tours, int generations, double percentageOfMutations, int size, CityWeight* cw [20][20])

Tours(int arr[]) : void

optimalCheckEliteOne(int arr[]);

optimalCheckEliteTwo(int arr[]);

optimalCheck(int arr[]) : double

```
1 runs (int arr[]) : void  
optimal Check Elite One (int arr[]);  
optimal Check Elite Two (int arr[]);  
optimal Check (int arr[]) : double  
Mutate() : void  
Generations (int arr[]) : void  
Random Num () : int  
get Elite One Weight() : double  
get Elite Two Weight() : double
```