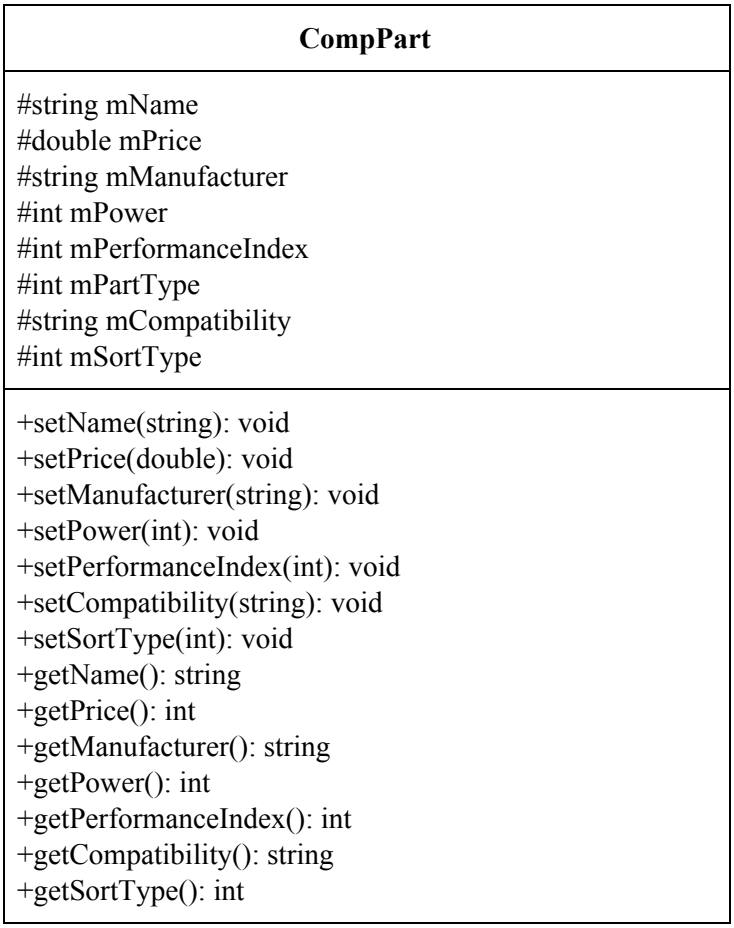
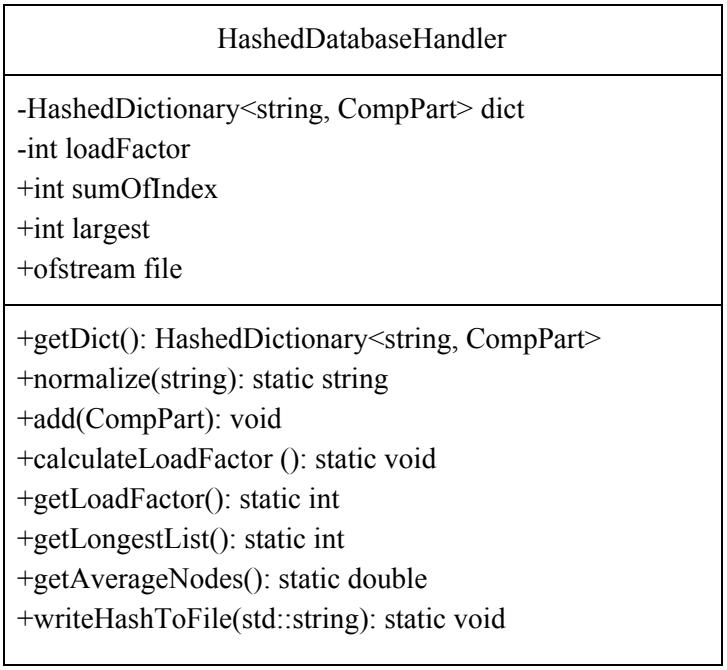
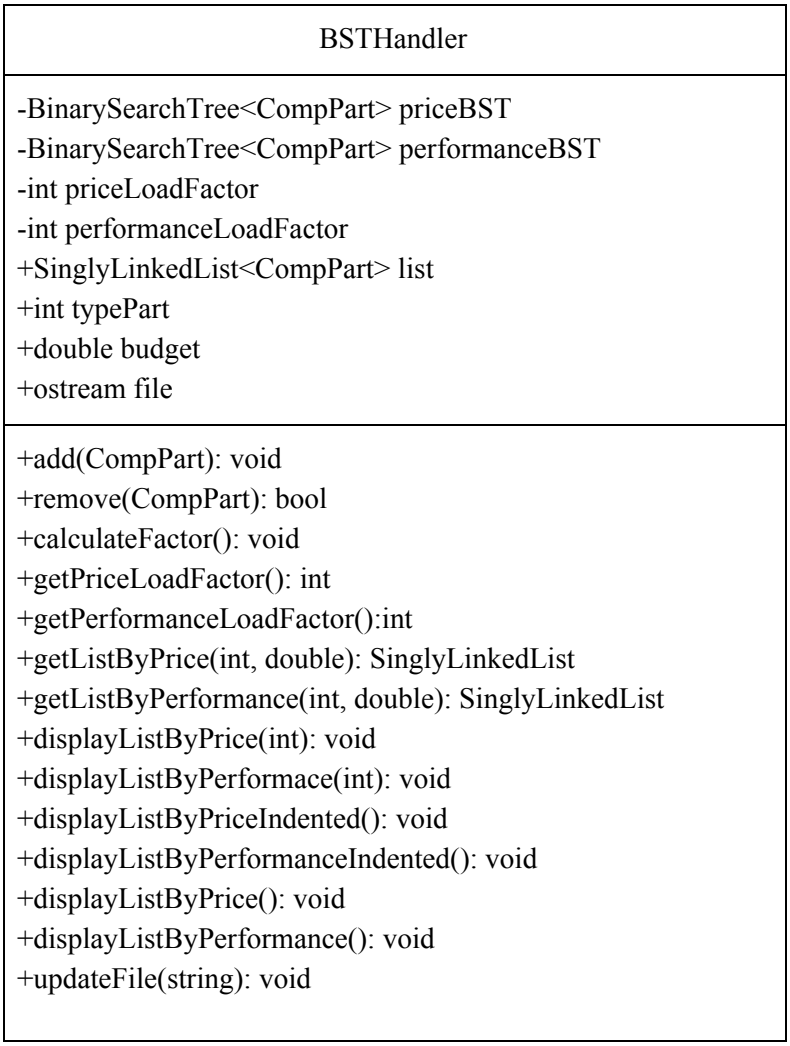
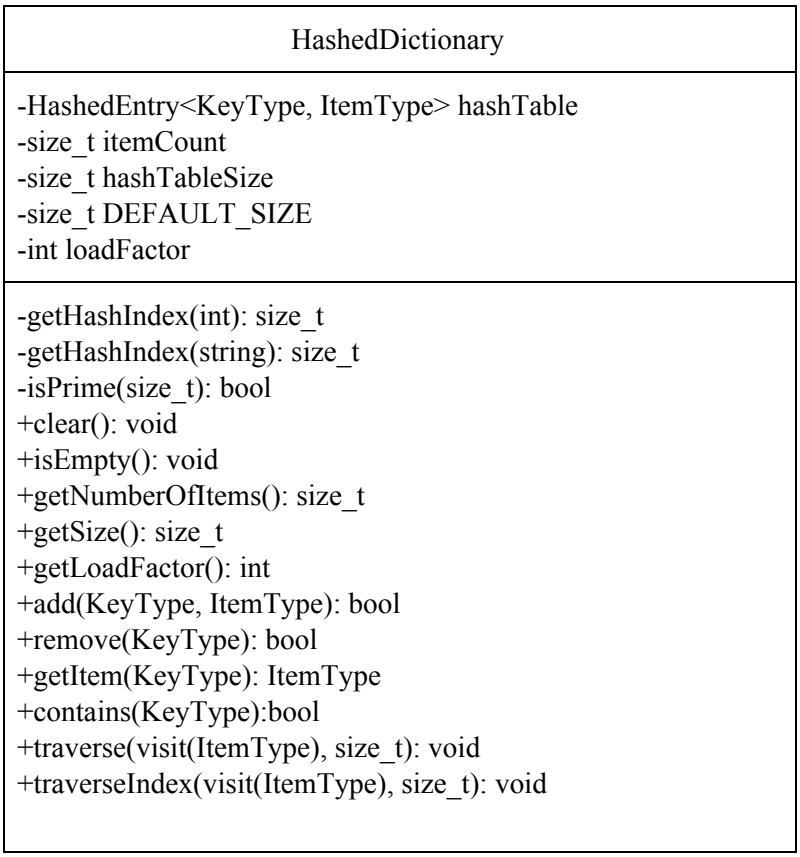

CompPart
#string mName
#double mPrice
#string mManufacturer
#int mPower
#int mPerformanceIndex
#int mPartType
#string mCompatibility
#int mSortType
+setName(string): void
+setPrice(double): void
+setManufacturer(string): void
+setPower(int): void
+setPerformanceIndex(int): void
+setCompatibility(string): void
+setSortType(int): void
+getName(): string
+getPrice(): int
+getManufacturer(): string
+getPower(): int
+getPerformanceIndex(): int
+getCompatibility(): string
+getSortType(): int


HashedDatabaseHandler
-HashedDictionary<string, CompPart> dict
-int loadFactor
+int sumOfIndex
+int largest
+ofstream file
+getDict(): HashedDictionary<string, CompPart>
+normalize(string): static string
+add(CompPart): void
+calculateLoadFactor (): static void
+getLoadFactor(): static int
+getLongestList(): static int
+getAverageNodes(): static double
+writeHashToFile(std::string): static void


BSTHandler
-BinarySearchTree<CompPart> priceBST
-BinarySearchTree<CompPart> performanceBST
-int priceLoadFactor
-int performanceLoadFactor
+SinglyLinkedList<CompPart> list
+int typePart
+double budget
+ostream file
+add(CompPart): void
+remove(CompPart): bool
+calculateFactor(): void
+getPriceLoadFactor(): int
+getPerformanceLoadFactor():int
+getListByPrice(int, double): SinglyLinkedList
+getListByPerformance(int, double): SinglyLinkedList
+displayListByPrice(int): void
+displayListByPerformace(int): void
+displayListByPriceIndented(): void
+displayListByPerformanceIndented(): void
+displayListByPrice(): void
+displayListByPerformance(): void
+updateFile(string): void


HashedDictionary
-HashedEntry<KeyType, ItemType> hashTable
-size_t itemCount
-size_t hashTableSize
-size_t DEFAULT_SIZE
-int loadFactor
-getHashIndex(int): size_t
-getHashIndex(string): size_t
-isPrime(size_t): bool
+clear(): void
+isEmpty(): void
+getNumberOfItems(): size_t
+getSize(): size_t
+getLoadFactor(): int
+add(KeyType, ItemType): bool
+remove(KeyType): bool
+getItem(KeyType): ItemType
+contains(KeyType):bool
+traverse(visit(ItemType), size_t): void
+traverseIndex(visit(ItemType), size_t): void