

Designing the Employee Management System

Object Oriented Concepts (OOPs)

OOP concept is very powerful coding techniques that have used as a design philosophy. Many programming languages support for OOP concept such as Java, C++, Python etc. Here, instead of traditional procedural programming language 'objects' have used to programming. Therefore, it is easy to understand the code well and these objects can reuse any time according to the OOP concepts. An object can define as a 'class' in many programming languages. Within this class all the features (variable) and functionalities of that object (methods) has defined. Therefore, an object can be defined as 'thing' that can perform a set of related activities. There are four main OOP concepts as describe as follows,

1. Inheritance – Inheritance can be defined as 'the ability of a new class to be created from an existing class by extending it'. Here the derived class called 'child class' or 'sub class' and the existing class called 'parent class' or 'base class' in object orientation. Therefore, the child class can inherit the parent class functionalities without recreating those functionalities again. Then this is some kind of reusability approach. Here the parent class is a general object which has common functionalities. On the other hand, child class can be considered as specialized object that has those common functionalities and some special functionalities of the object.
2. Polymorphism – This concept can be defined as 'ability to request the same operation to perform different types of things. In OOP, this concept can be divided into different such that method overloading, operator overloading and method overriding. Here method overloading is the ability to define several methods using the same name. Operator overloading is a specific case of polymorphism that can use the normal operators as polymorphic functions to get different behaviors with different arguments. Method overriding allows a child class to override a specific implementation of the method that is already defined in parent class.
3. Encapsulation – The basic idea of the encapsulation is hide the functionality of the class while allowing other class to use that functionality without actually knowing the implementation. Basically, user can access class data using setter and getter which are the standard methods.
4. Abstraction - The main goal of the abstraction is handle complexity by hiding unnecessary details from the user. Basically, abstraction can be used to define the abstract class.

Main features in C++

There are some important features in C++ when comparing with other OOP languages. Therefore, many companies such as EDA companies use this language as an advance version of the C language. Some important features have described in following points.

1. Multiple inheritance – C++ can inherit from more than one class to derive a subclass. This feature not support for other OOP programming languages.
2. C++ can be used to do low level programming. Therefore, this can be used to design games, operating system kernels, drivers etc.
3. This can be used as a structural as well as OOP programming language. Because this language is combination of C and OOP concepts.
4. The compilation and execution time is fast
5. Support accessing memory via pointers and support for recursion functions.

6. Support for friend classes and virtual

Design Planning

Before implementing the software, it is necessary to have a design planning. Otherwise it may be faced many difficulties when implementing an optimized software. Therefore, before coding the EMS software a class diagram was created with relevant associations. In the class diagram, relevant functions and attributes were defined according to the given specifications and presented it to the senior software engineers. During this presentation, they were highlighted important facts. Then the class diagram was improved according to the given instructions. Therefore, this class diagram provides the connectivity of each class and usage of OOP concepts when creating the classes. Specially, when designing the class diagram, some security features were implemented to limit the accessing of some data. Therefore, 'singleton' antipattern was used to implement this security feature. As shown in figure 5.1 it can be illustrated the class diagram that was used for creating the EMS.

Employee Management System (EMP) and Specifications

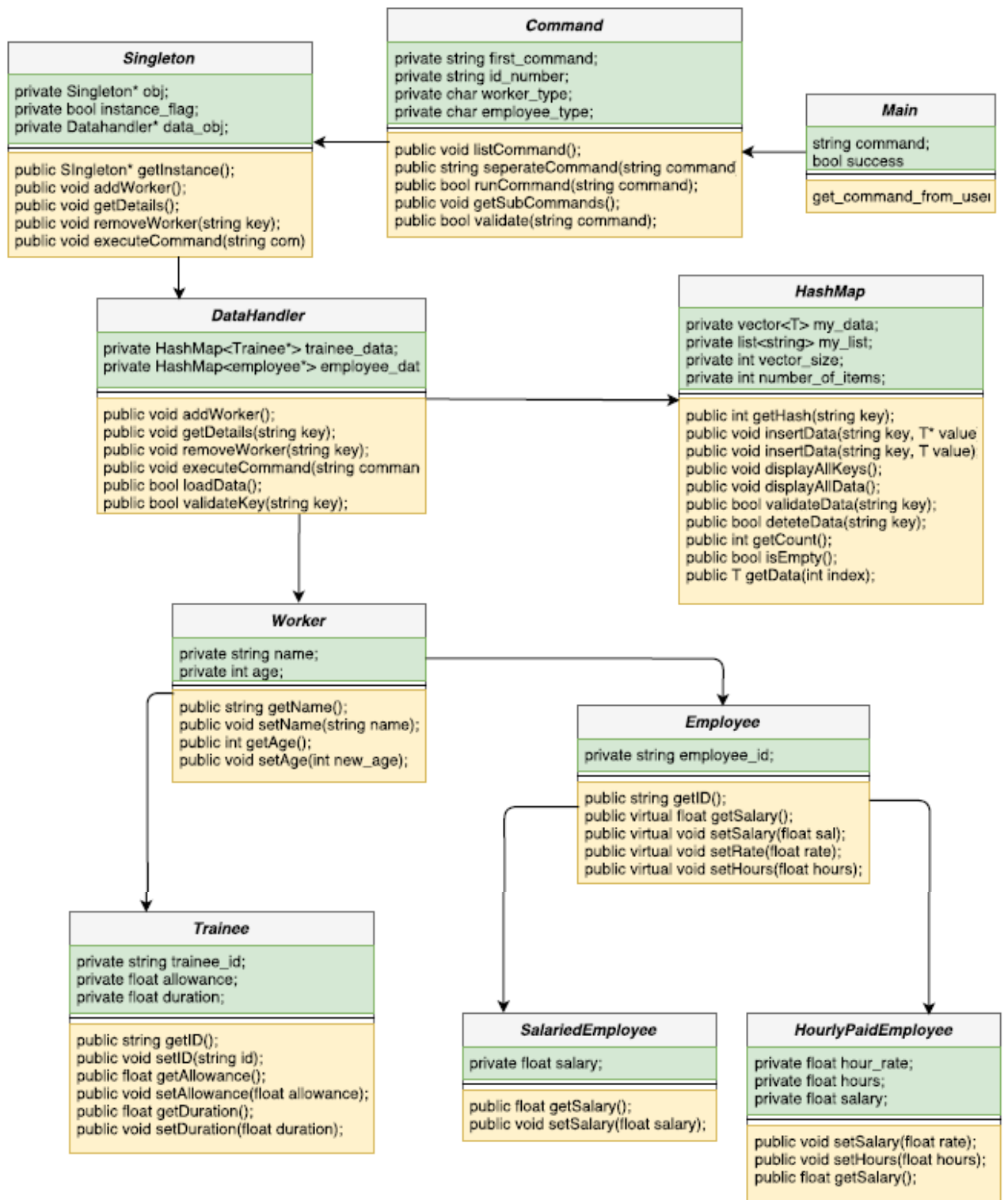
For the EMS it should be followed the class diagram that was designed already. Here EMP should have following features as given specifications.

1. Implement a suitable data structure using C++ for storing employee data
2. The EMS should return the relevant employee data with minimum time. Therefore, the data structure should be implemented with low time and space complexity.
3. Concern about the security when accessing data
4. Use OOP concepts for reusing the objects and its functionalities.
5. Should have features such as adding, removing and getting monthly payment for all kinds of employees etc.
6. Improve the user friendliness of the command interface
7. Coding with proper standard such as following proper naming for functions and variables with better documentation.

After coding the software according to given specifications, a bash script was implemented to execute the program. Then the all behavior of the code was checked for specified features.

Debugging and final outcome

After checking the all behaviors of the software, a problem was found when invoking the remove and get_details for particular employee ID. Therefore, the 'gdb' debugger tool on Linux platform was used to debug the entire software. Using this tool, a memory issue was found out. Then the 'Valgrind' tool was used to find the exact memory leaks of the software. Actually, the exact problem was reallocating the allocated memory when storing the employees in the 'hashmap' data structure. After implementing the proper re-allocating of the allocated memory, all the features have functioned without any undefined behavior. Finally, the behavior of the software was presented to the senior software engineers in the company and got the instruction to further development of coding skills by following the standard practices in software engineering. Figure 5.2 illustrates the result of 'help' command that lists all features and particular commands (command manual).



```
##### Well Come to the Employee Management System #####
```

```
Please enter the command here: help
```

```
##### Command Manual #####
```

```
add          : Add a worker to the data base
list_all     : List the all data that have already stored
list_monthly_pay : List monthly pay for all employees
get_total_pay : List monthly total pay of workers
get_details <id> : Get details for particular ID
remove <id>    : Remove particular ID from the database
exit         : Exit from the software
help         : Display command manual
```

```
#####
```

```
Please enter the command here: █
```