

# **DATA STRUCTURE GROUP PROJECT**

## **Introduction**

A database is an organised collection of data, generally stored and accessed electronically from a computer system. It stores persistent (permanent) or volatile (temporary) data more appropriately instead of just dumping everything into a vast array.

Usually, when using a database in a system, the programmer will take account of the database models before using them. Some of the most common database models are the relational and non-relational models. Relational databases became dominant in the 1980s. These model data are rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, referred to as NoSQL because they use different query languages.

## **Task**

Create a simple database engine that can store at least these data types into one database:

- Numeric data (int, long, byte, double, float, etc.)
- String data (string, char)
- Collections (arrays of the types above)

The database engine must be in the form of non-relational, which means that when storing the data, it must not be in a tabular form.

## **Requirements**

1. Must be able to store data with different datatypes successfully.
2. Must be able to retrieve one, or all the data.
3. Must be able to display the stored data
4. Must be able to delete one data
5. Must be able to clear all the data stored

```

run:
My Simple Database Engine
Please enter command:
- insert: Insert new data
- display: Display all data
- get: Get a specific data
- delete: Delete a specific data
- clear: Clear all data stored
Command: insert

Please enter a datatype ('str' for String, 'c' for character, 'num' for numerical, 'X' for exit): str
Please enter the data: My Simple Database
Please enter the index: myMessage
[Database] Inserting....
[Database] Done!
Please enter a datatype ('str' for String, 'c' for character, 'num' for numerical, 'X' for exit): num
Please enter the data: 21
Please enter the index: myAge
[Database] Inserting....
[Database] Done!
Please enter a datatype ('str' for String, 'c' for character, 'num' for numerical, 'X' for exit): c
Please enter the data: 1
Please enter the index: myGender
Please enter a datatype ('str' for String, 'c' for character, 'num' for numerical, 'X' for exit): num
Please enter the data: duapuluhsatu
Please enter the index: myFriendsAge
[Database] Error! Not a number! Please try again!
Please enter a datatype ('str' for String, 'c' for character, 'num' for numerical, 'X' for exit): num
Please enter the data: 1.85
Please enter the index: myHeight
[Database] Inserting....
[Database] Done!
Please enter a datatype ('str' for String, 'c' for character, 'num' for numerical, 'X' for exit): X

My Simple Database Engine
Please enter command:
- insert: Insert new data
- display: Display all data
- get: Get a specific data
- delete: Delete a specific data
- clear: Clear all data stored
Command: display

Display all data

```

index	type	value
myMessage	string	My Simple Database
myAge	number	21
myGender	character	1
myHeight	number	1.85

```

BUILD SUCCESSFUL (total time: 0 seconds)

```

Note: Your outputs may be different than this. This is not the final output. This is only a simulation of the program of what it should do.

For this project, it is advisable for you to use a minimum of one of these data structures or techniques;

1. HashMap
2. Generics
3. Linked list
4. Inheritance
5. Polymorphism

It is required for you to fulfil the requirements first. No bonus marks will be given if you fail to accomplish all of them, even though you managed to do extras in this project.

## Guides

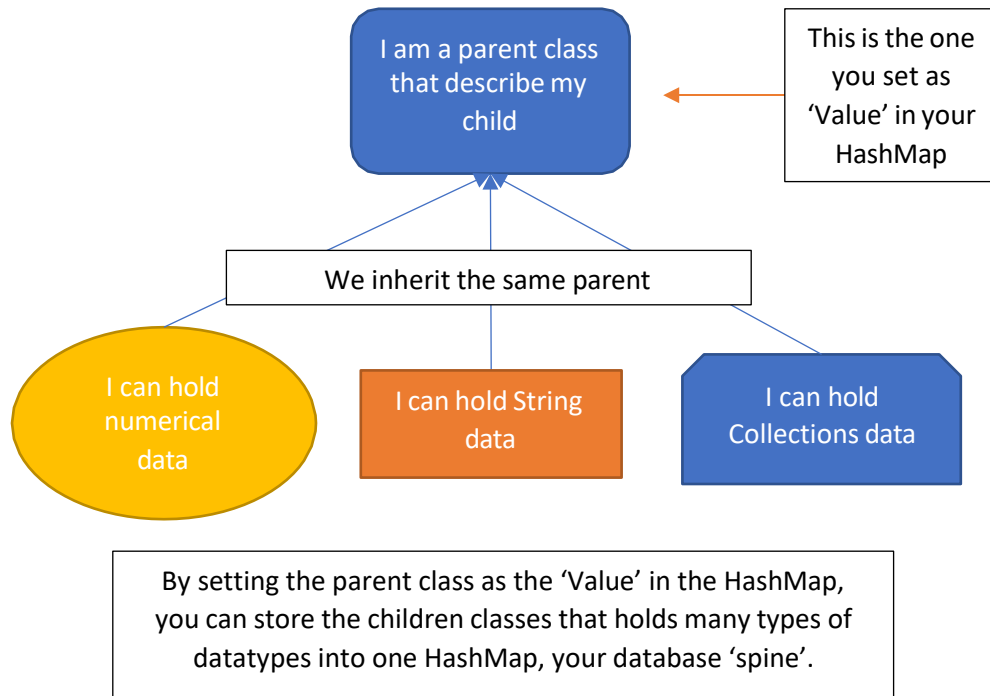
Here are some guidelines on how to design your simple database engine:

1. Use HashMap. You can use simple or generic arrays, but by using HashMap, it is much easier to make the “spine” of the database. It is also very versatile for you to use it. Refer to the code snippets below:

```
/*  
  
    databaseSpine is the main collections for storing all the data that  
    will be inserted into your database engine later.  
  
    When declaring HashMap, please take note the notation of the class;  
  
    HashMap<Key, Value>  
  
    - Key is the datatype which you want to use as the 'index' of the item  
    - Value is the datatype which you want to store into the list  
  
*/  
  
HashMap<String, Object> databaseSpine = new HashMap<>();  
  
// This is how you store an item into the 'spine'  
databaseSpine.put("tableName1", new Object());  
  
// This is how you get an item from the 'spine'  
  
Object tableFromSpine = databaseSpine.get("tableName1");
```

2. Apply the inheritance concept in this project. This concept is fundamental if you want to be able to store many data types into one 'spine'.

Hint: the 'Value' in the HashMap is a class that inherits a parent class that describes the child classes as classes that hold data. Refer to the figure below.



That is how you store many data types into one HashMap.

3. That is all. Good luck! 😊

### Bonus

Bonus marks will be given for these extra tasks:

1. Data in the "spine" can be stored on the disk.
2. Ability to create a table in your database engine.
3. More datatypes besides the listed datatypes.
4. Graphical User Interface.
5. Many more other extras

## **Project Marks / Rubrics**

The following marks serve as a guideline:

### **Mastery of data structure: 20%**

- Showcase structure development – data, methods, constructors, and any required components to successfully implement the structure.

### **Fulfilment of question's requirements: 15%**

- Showcase workable traversing/searching of structure – 5%
- Showcase workable element/data editing (adding, deleting, sorting, etc.) – 5%
- Showcase correct result – 5%

### **Bonus marks (optimisation, wow factor): 10%**

- GUI has some merit.
- Less popular project has some merit.

### **Submission of 5-minute Video Presentation: 5%**

- Showcase mastery of data structure.
- Showcase project requirements achievement.

Submission of Project report – no marks but required for formalities. Can be a 2-page report describing your group member details, 1 paragraph describing project selection (why you chose the project), and 1 paragraph reflecting the challenges faced during the development. The report can be Word Doc, PDF or PPT slides (as long you can submit through Spectrum).

Project Deadline – will be published on Spectrum.