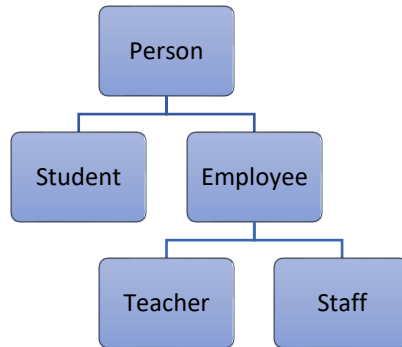


Lab 3

JavaScript Classes and Prototypes

Task 1

Implement JavaScript function-based classes (NOT ES6 CLASSES) for the following scenario:



Student Management System containing Person, Student, Employee, Teacher, Staff and Courses classes.

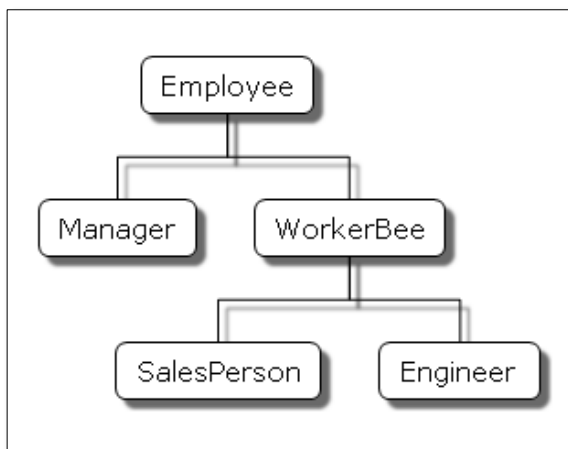
- Student and Employee classes are the derived class of Person
- Teacher and Staff are derived classes of Employee
- Person class contains 3-4 common fields of Student and Employee. Initialize the fields with the default value
- Employee class contains 3-4 common fields of Teacher and Staff (such as department, designation, salary, etc)
- Student, Teacher, and Staff contains the fields specific to their role
- Courses have an aggregation type of association with Student and Teacher classes

Finally, create two objects of Student, Teacher, and Staff each and print their information through created objects on the console.

Task 2:

Implement the following Classes with required properties using ES6 Classes.

Also, implement Getter and Setter methods in all classes for every instance variable.



- `Employee` has the properties `name` (whose value defaults to the empty string) and `dept` (whose value defaults to "general").
- `Manager` is based on `Employee`. It adds the `reports` property (whose value defaults to an empty array, intended to have an array of `Employee` objects as its value).
- `WorkerBee` is also based on `Employee`. It adds the `projects` property (whose value defaults to an empty array, intended to have an array of strings as its value).
- `SalesPerson` is based on `WorkerBee`. It adds the `quota` property (whose value defaults to 100). It also overrides the `dept` property with the value "sales", indicating that all salespersons are in the same department.
- `Engineer` is based on `WorkerBee`. It adds the `machine` property (whose value defaults to the empty string) and also overrides the `dept` property with the value "engineering".

Task 3: React Native & Expo CLI

Part (i) Snack

- Go to snack.expo.dev and create an account.
- Download the Expo Go application and log in to your account there too.
- Write a "Hello World" program for functional and class components separately and save these snacks with the names "Functional component" and "Class component".

Part (ii) React Native CLI

- Create an RN application with the name "First application".
 - Use this command to create the application "`npx react-native init First Application`"
- Open the project using VS code and go to the **App.js** file
- Inside the file, write a functional component with a "Hello World" message.