1) Override showDetails() method in the sub classes to include the details of additional fields. Use super key word to call the methods of super class if required.

2) Create an array of Job class and store Objects of Job, PriorityJob, and MultiOwnerJob classes in the array.

3) Using a single for loop try traverse the above array and call the method showDetails() on all the objects of the array. Understand the concept of runtime polymorphism.

4) Check is it possible to call all the methods of PriorityJob and PriorityJob while traversing the array. If not use typecasting to achieve

the above task.

5) Complete the below tasks

a. Create Interfaces
b. Create inheritance by extending other interfaces
c. Creating class by extending another class and implementing more than 1 interface
d. Create a reference variable of an interface.
e. Create a class implementing above interface.
f. Store the object created in step e in the reference variable created in step d.
g. Call the methods by using interface reference

6) A pizza delivery outlet uses a software for maintaining information about their pizzas. Structure of the pizza class is given below

Class Pizza {

String pizzaName // unique field

String description;

Int sizeIncms

String majorIngredientOne

String majorIngredientTwo

String majorIngredientThree

Int weight

float price

public void preparation()

{

// this method displays the procedure for preparation

// identify other necessary methods and properties

}

Develop a layered application so that view layer is responsible for user interaction and is only responsible for accepting and displaying details from /to user. View layer will communicate with storage layer for all CRUD application. Class structures, methojd information and responsibilities of each class are given below.

Create a main class to test your application.

View Layer	**Storage Layer**
CustomerView(Class) AddPizzaDetailsandStrore() displayPizzaDetailsbyName() Responsbility: This class accept the details from the user and call the methods on storage layer. Call PizzaStorageFactory.getpizzaStrorage() to get the Storage implementation object.	**PizzaStore(Interface)** void addNewPizza(Pizza e) Pizza getPizzaByName(String pizzaname) Pizza[] getPizzaByCountBySize(int size)
	Responsibility: This class implements PizzaStore Create an array to store the pizza objects and provide proper implementation to the methods in the interface

	PizzaStoreFactory(Class) Public static PizzaStrore getpizzaStore() **Responsibility:** This method is used to get the object of PizzaStore implementation. View layer should call this method to get the reference of the PizzaStroreImpl