ADF-I Assignment 9: Abstract, Interface

Part I:

Write a Java application - Inventory System - to manage the list of televisions with the specification as follows:

1	Creates an **abstract class** named **Product** in package **Goods.** - Protected Fields **id, name** - Public constructor to initialise the above fields. - Method: - Protected void **accept()** : allow user input data into data fields. - Public abstract void **printInfo()** : **abtract method** used to print details of an product.
2	Create an interface **ITax** in package **Goods** , consists of: - Field **VAT_TAX_PERCENT = 0.1 f** - Method: public float **getCost():** returns the cost of a product after TAX.
3	Create class **Television** **derives from** **Product** and **implements** **ITax**, in package **Electronics.** - Fields: p**price, QoH** (quantity on hand) and **brand.** - Constructors to initialise the all fields. - Override methods: **protected void accept() :** allow user to input additional details of a television invoke method accept() of super class. **public String toString() :** return a string presenting all the details of a product as follows: *id, name, price, QoH, cost, amount* (=cost*QoH, cost: price after TAX)
4	Create class **TelevisionCatalog** in package **Electronics** for managing a collection of Televisions: - Fields: **[max, count]** int**, tvList – array of** Television**.** - Default constructor to initialise the all the fields. - Methods: - **Public void add() -** add a new television into array - **Public void searchByBrand() -** search televisions belong a brand name accepted by user. - **Public void displayAll() -** display all televisions. - **Public void displayHighValue()** – display televisions with the price above 500.
5	Create **main** class **Inventory** in package **Application** that allows user to manage the televisons accepted into system through the menu system as follows: 1. Add a new television 2. Search televisions by brand 3. Display all televisions 4. Display high-valued televisions 5. Exit

Part II (+)

Write a Java application – Payroll System - to manage the payroll system of employees in a company:

1	Creates an **abstract class** named **Employee** in package **data.** - Private Fields [**ID, name]:** string**, [workedDays, salary]:** int - Constructors to initialize the above fields. - Method: - Proctected void **accept()** : allow user input data into data fields. validation: ID, name is not null, workedDays, salary > 0. ID *is not duplicate***.** - Public *abtract* void **printInfo()** : used to print the pay slip of a*n*y employee.
2	Create an interface **ISalary** in package **data** , consists of: - Method: public float **getAllowance():** calculates and returns the allowance of an employee. public float **getIncome():** calculates and returns the actual salary of an employee
3	Create an interface **ITaxable** in package **data** , consists of: - Field **INCOME_TAX_PERC = 0.3 f** - Method: public float **getTax():** calculates and returns the income tax of an employee.
4	Create class **Worker derives from Employee** and **implements** interfaces **IAllowance, ITaxable**, in package **data,** consists of: - Private Field **[overtime]:** int - Constructors to initialize the all fields. - Override methods: - proctected void **accept() :** allow user to input all details for a worker. (Hint: invoke super.accept()) - validation: overtime >= 0 - public float **getTax():** monthly sal = salary*(worked days + overtime*2) /26; tax = 0 if mothly sal < 400 else tax = monthly sal * income tax percent - public float **getAllowance():** 0 - public float **getIncome():** monthly sal – tax - public void **printInfo()** : print the pay slip of a worker - public String **toString() :** return a string presenting all details of a worker .
5	Create class **Engineer derives from Employee** and **implements** interfaces **IAllowance, ITaxable**, in package **data,** consists of: - Private Field **[level]:** int - Constructors to initialize the all fields. - Override methods: - proctected void **accept() :** allow user to input all details for an egineer. (Hint: invoke super.accept()) - validation: level >= 1 and <= 4

	- public float **getAllowance():** level = 1 => allownce= 400, level = 2 => allownce= 600 level = 3 => allownce= 1000,level = 4 => allownce= 2000 - public float **getTax():** monthly sal = salary * worked days /24 + allowance ; tax = monthly sal * income tax percent - public float **getIncome():** monthly sal – tax - public void **printInfo()** : print the pay slip of a worker - public String **toString() :** return a string presenting all details of an engineer
6	Create class **EmployeeCatalog** in package **data** for managing a collection of Employee - Fields: **eList –** array list of employees**,** - No-arg constructor to initialize the all the fields. - Methods: - **addWorker() -** add a new worker - **addEngineer() -** add a new engineer - **display() -** display all employees. - **display(**boolean **isWorker) –** display a list of workers if **isWorker** is true, otherwise display a list of all engineers – (Hint – using **instanceof** operator) - **display(**string **name) -** search and display the pay slip of employees having name accepted by user.
7	Create java **main** class **PayrollSystem** in package **Application** with the menu system as follows: 1. Add a new worker 2. Add a new engineer 3. Display all employees 4. Display slist of workers 5. Display list of engineers 6. Display the pay slip of any employee by name. 7. Exit