HOMEWORK

Task 1

Write the "**Product**" class to show the following output

Note: Make sure to use proper ***Encapsulation concepts*** **for the setter & getter methods. All the attributes should have Private access.**

Driver Code	Output

public class ProductTester{
  public static void main(String[] args) {
  System.out.println("< --------1--------->");
  Product product1 = new Product();
  product1.displayInfo();
  System.out.println("< --------2--------->");
  Product product2 = new Product("Laptop", 1200.00);
  product2.setQuantity(10);
  product2.displayInfo(true);
  System.out.println("< --------3--------->");
  System.out.println("Retrieved Price: $" + product2.getPrice());
  System.out.println("Retrieved Quantity: " + product2.getQuantity());
  }
}
	< --------1--------->
	Product Name: Unknown
	Price: $0.0
	< --------2--------->Product Name: Laptop
	Price: $1200.0
	Quantity: 10
	< --------3--------->Retrieved Price: $1200.0
	Retrieved Quantity: 10

Task 2

Design the Company and Employee classes so that the Tester1 class produces the given outputs. [All attributes of Employee class should be **Private**]
Restriction: Company class can't have more than 1 array.

Driver Code	Output

public class Tester1{
  public static void main(String args[]){
  Employee e1 = new Employee();
  Employee e2 = new Employee("Alif", 34, "Fulltime");
  System.out.println("1------------");
  Company c1 = new Company();
  c1.details();
  System.out.println("2------------");
  Employee e3 = new Employee("Akter", 36,"Part-time");
  Employee e4 = new Employee("Ria", 38, "Fulltime");
  System.out.println("3------------");
  c1.addEmployee(e2);
  c1.addEmployee(e3);
  System.out.println("4------------");
  c1.details();
  System.out.println("5------------");
  c1.addEmployee(e4);
  c1.addEmployee(e1);
  System.out.println("6------------");
  c1.details();
  System.out.println("7------------");
  c1.removeEmployee(e4);
  System.out.println("6------------");
  c1.details();
  }
}
	A default employee has been created
	1------------
	Company Name: ABC Company
	Total Employee: 0
	Fulltime Employees:
	Part-Time Employees:
	2------------
	3------------
	Alif has joined the company
	Akter has joined the company
	4------------
	Company Name: ABC Company
	Total Employee: 2
	Fulltime Employees:
	Name: Alif, ID: 34
	Part-Time Employees:
	Name: Akter, ID: 36
	5------------
	Ria has joined the company
	No more vacancy
	6------------
	Company Name: ABC Company
	Total Employee: 3
	Fulltime Employees:
	Name: Alif, ID: 34
	Name: Ria, ID: 38
	Part-Time Employees:
	Name: Akter, ID: 36
	7------------
	Ria has left the company
	6------------
	Company Name: ABC Company
	Total Employee: 2
	Fulltime Employees:
	Name: Alif, ID: 34
	Part-Time Employees:
	Name: Akter, ID: 36

Task 3

Please write the **Student** and **Department** class with the necessary properties so that the provided driver code generates the output given below. The id of the Student class will be **private**. For simplicity, assume that a department can add a maximum of 5 students.

Driver Code

```
public class Tester1 {
 public static void main(String[] args) {
  Student s1 = new Student("Akib", 10, 3.29);
  Student s2 = new Student("Reza", 15, 3.45);
  Student s3 = new Student("Kabir", 20,4.0);
  System.out.println("1==========");
  Department cse = new Department("CSE");
  cse.findStudent(-100);
  System.out.println("2==========");
  cse.addStudent(s1, s2, s3);
  System.out.println("3==========");
  cse.details();
  System.out.println("4==========");
  cse.findStudent(15);
  System.out.println("5==========");
  Student s4 = new Student("Nakib", 15,3.22);
  cse.addStudent(s4);
  System.out.println("6==========");
  s4.setId(25);
  cse.addStudent(s4);
  System.out.println("7==========");
  cse.details();
  System.out.println("8==========");
  Student s5 = new Student("Sakib", 30,2.29);
  cse.addStudent(s5);
  System.out.println("9==========");
  cse.details();
 }
}
```

Output

```
1==========
Student with this ID doesn't exist, Please
give a valid ID
2==========
Welcome to CSE department, Akib
Welcome to CSE department, Reza
Welcome to CSE department, Kabir
3==========
Department Name: CSE
Number of student:3
Details of the students:
Student name: Akib, ID: 10, cgpa: 3.29
Student name: Reza, ID: 15, cgpa: 3.45
Student name: Kabir, ID: 20, cgpa: 4.0
4==========
Student info:
Student Name: Reza
ID: 15
CGPA:  3.45
5==========
Student with the same ID already exists,
Please try with another ID
6==========
Welcome to CSE department, Nakib
7==========
Department Name: CSE
Number of student:4
Details of the students:
Student name: Akib, ID: 10, cgpa: 3.29
Student name: Reza, ID: 15, cgpa: 3.45
Student name: Kabir, ID: 20, cgpa: 4.0
Student name: Nakib, ID: 25, cgpa: 3.22
8==========
Welcome to CSE department, Sakib
9==========
Department Name: CSE
Number of student:5
Details of the students:
Student name: Akib, ID: 10, cgpa: 3.29
Student name: Reza, ID: 15, cgpa: 3.45
Student name: Kabir, ID: 20, cgpa: 4.0
Student name: Nakib, ID: 25, cgpa: 3.22
Student name: Sakib, ID: 30, cgpa: 2.29
```

Task 4

Spaceship: This class represents a spaceship. Each spaceship has a **name** and a **capacity** (the maximum weight it can carry).
Cargo: This class represents a piece of cargo. Each cargo item has a **name** and a **weight**. Both attributes should be **private** which means they cannot be accessed directly from outside of the class.
A Spaceship contains Cargo. That means each spaceship can carry multiple cargo items, but the total weight of the cargo cannot exceed the spaceship's capacity. Also, the maximum number of cargo items is 100. Your task is to design the **Spaceship** and **Cargo** class with necessary properties so that the given output is produced for the provided driver code.

Driver Code	Output

public class Tester2 {
 public static void main(String[] args) {
  Spaceship falcon = new Spaceship("Falcon", 50000);
  Spaceship apollo = new Spaceship("Apollo", 100000);
  Spaceship enterprise = new Spaceship("Enterprise",220000);
  System.out.println("1.==========");
  Cargo gold = new Cargo("Gold", 20000);
  Cargo platinum = new Cargo("Platinum", 25000);
  Cargo dilithium = new Cargo("Dilithium", 50000);
  Cargo trilithium = new Cargo("Trilithium", 70000);
  Cargo neutronium = new Cargo("Neutronium", 80000);
  System.out.println("2.==========");
  falcon.loadCargo(gold);
  falcon.loadCargo(platinum);
  falcon.displayDetails();
  System.out.println("3.==========");
  apollo.loadCargo(gold);
  apollo.displayDetails();
  System.out.println("4.==========");
  falcon.loadCargo(neutronium);
  System.out.println("5.==========");
  enterprise.loadCargo(dilithium);
  enterprise.loadCargo(trilithium);
  enterprise.loadCargo(neutronium);
  enterprise.displayDetails();
 }
}
	1.==========
	2.==========
	Spaceship Name: Falcon
	Capacity: 50000
	Current Cargo Weight: 45000
	Cargo:Gold Platinum
	3.==========
	Spaceship Name: Apollo
	Capacity: 100000
	Current Cargo Weight: 20000
	Cargo:Gold
	4.==========
	Warning: Unable to load Neutronium inside Falcon. Exceeds capacity by 75000.
	5.==========
	Spaceship Name: Enterprise
	Capacity: 220000
	Current Cargo Weight: 200000
	Cargo:Dilithium Trilithium Neutronium