Class Employee

```
public abstract class Employee {
	private String name;
	private int id;
	public Employee(String name, int id) {
		this.name = name;
		this.id = id;
	}
	public Employee(Employee e){
		this.name = e.name;
		this.id = e.id;
	}
	public String getName() {
		return name;
	}
	public int getId() {
		return id;
	}
	public void display(){
		System.out.println("Employee name:  " + name);
		System.out.println("Employee id:  " + id);
	}
	public abstract double calculatePay();

}
```

Class FullTime

```
public class FullTime extends Employee{
	private double salary;
	public FullTime(String name, int id, double salary) {
		super(name, id);
		this.salary = salary;
	}
	public FullTime(FullTime ft){
		super(ft);
		this.salary = ft.salary;
	}
	public void display(){
		super.display();
		System.out.println("Salary:  " + salary);
	}
	public double calculatePay(){
		return salary - salary*0.09;
	}
	public double getSalary() { return salary; }
}
```

Class PartTime

```java
public class PartTime extends Employee{
	 private int nbWorkHours;
	 private int rate;
	 public PartTime(String name, int id, int nbWorkHours, int rate) {
		super(name, id);
		this.nbWorkHours = nbWorkHours;
		this.rate = rate;
	}
	public PartTime(PartTime pt){
		super(pt);
		this.nbWorkHours = pt.nbWorkHours;
		this.rate = pt.rate;
	}
	public void display(){
		super.display();
		System.out.println("Number of work hours: " + nbWorkHours);
		System.out.println("Rate of each hour: " + rate);
	}
	public double calculatePay(){
		return nbWorkHours * 4 * rate;
	}
	public int getNbWorkHours() { return nbWorkHours; }
	public int getRate() {return rate;}
}
```

Class Company

```java
public class Company {
      private String name;
      private Employee[] arrEmployee;
      private int nbEmployee;

public Company(String name, int size) throws NegativeArraySizeException{
            if(size < 0) throw new NegativeArraySizeException();
            this.name = name; arrEmployee = new Employee[size];
            nbEmployee = 0;
      }
      public void displayAll(){
            for(int i = 0; i < nbEmployee; i++) arrEmployee[i].display();
      }
      public void addEmployee(Employee e) throws IllegalStateException{
            if(nbEmployee >= arrEmployee.length)
                  throw new IllegalStateException("Array is full!");
            if(e instanceof PartTime)
                  arrEmployee[nbEmployee++] = new PartTime((PartTime) e);
            else arrEmployee[nbEmployee++] = new FullTime((FullTime) e);
      }
private int searchName(String name){
      for(int i = 0; i < nbEmployee; i++)
            if(arrEmployee[i].getName().equalsIgnoreCase(name)) return i;
      return -1;
      }
public void deleteEmployee(String name) throws IndexOutOfBoundsException{
            int index = searchName(name);
            if(index == -1) throw new IndexOutOfBoundsException();
            arrEmployee[index] = arrEmployee[nbEmployee-1];
            arrEmployee[nbEmployee-1] = null; nbEmployee--;
      }
public double getYearlyPay(String name) throws IndexOutOfBoundsException{
            int index = searchName(name);
            if(index == -1) throw new IndexOutOfBoundsException();
            return arrEmployee[index].calculatePay() * 12;
      }
      public double calAvgPayForPartTime() throws ArithmeticException{
            int countPT = 0; double sum = 0;
            for(int i = 0; i < nbEmployee; i++)
                  if(arrEmployee[i] instanceof PartTime){
                        countPT++;
                        sum += arrEmployee[i].calculatePay();
                  }
            if(countPT == 0) throw new ArithmeticException("/ by zero");
            return sum/countPT;
      }
}
```

Class Test

```
public class test {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		PartTime e1 = new PartTime("Ahmad", 111, 6, 150);
		PartTime e2 = new PartTime("Omar", 222, 10, 200);
		PartTime e3 = new PartTime("Khalid", 333, 9, 150);
		FullTime e4 = new FullTime("Mohammed", 444, 5000);
		FullTime e5 = new FullTime("Ali", 555, 10000);
		try{
			Company c = new Company("KSU", 4);
			try{
				c.addEmployee(e1);
				c.addEmployee(e2);
				c.addEmployee(e3);
				c.addEmployee(e4);
				c.addEmployee(e5);
			}catch(IllegalStateException e){
				System.out.println(e.getMessage());
			}
			try{
				c.deleteEmployee("Ahmad");
				System.out.println("Delete done");
			}catch(IndexOutOfBoundsException e){
				System.out.println(e);
			}
			try{
	System.out.println("Yearly pay of Omar: " + c.getYearlyPay("Omar"));
			}catch(IndexOutOfBoundsException e){
				System.out.println(e);
			}
			try{
	System.out.println("Average PartTime: " + c.calAvgPayForPartTime());
			}catch(ArithmeticException e){
				System.out.println(e);
			}
		}catch(NegativeArraySizeException e){
			e.printStackTrace();
		}
	}
}
```