```java
/**
* This employee class is an abstract class that
* implements EmployeeInfo and is the parent class for Staff and Faculty
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
public abstract class Employee implements EmployeeInfo {
    //initialize variables used by all employees
    String last;
    String first;
    String id;
    //default constructor
    //sets variables to blank
    public Employee(){
        last = "BLANK";
        first = "BLANK";
        id = "BLANK";
    }
    //constructor that takes in a last name first name and an id
    public Employee(String last, String first, String id){
        this.last = last;
        this.first = first;
        this.id = id;
    }
    //setters and getters for variables: first, last, and id
    public String getLastName(){
        return last;
    }
    public void setLastName(String newLast){
        last = newLast;
    }
    public String getFirstName(){
        return first;
    }
    public void setFirstName(String newFirst){
        first = newFirst;
    }
    public String getId(){
        return id;
    }
    public void setId(String newId){
        id = newId;
    }
    //abstract method that will be filled in in subclasses
    public abstract double monthlyEarning();

    //to String that prints out all the variables of employee: first name, last name, and ID
    @Override
    public String toString() {
        return "Employee{" +
                "last='" + last + '\'' +
                ", first='" + first + '\'' +
                ", id='" + id + '\'' +
                '}';
    }

}
```

```
/**
* Staff is a subclass of Employee that inherits its variables(first, last, id) and
* has a new variable called hourly rate
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
public class Staff extends Employee{
    double hourly_rate;
    //default constructor sets hourlyrte to 0
    public Staff(){
        super();
        hourly_rate = 0;
    }
    //constructor that takes in lasat,first,id and hourly rate as paramaters
    public Staff(String last, String first, String id, double hourly_rate){
        super(last, first, id);//gets last,first, and id from fatherclass
        this.last = last;
        this. first = first;
        this.id = id;
        this.hourly_rate = hourly_rate;
    }
    //override the monthlyEarning and returns the hourly rate times the STAFF_MONTHLY_HOURS_WORKED
    //which is gotten from the interface EmployeeInfo
    @Override
    public double monthlyEarning() {
        return hourly_rate * STAFF_MONTHLY_HOURS_WORKED;
    }
    //to String prints out all the variables from employee as well as the monthly earning
    @Override
    public String toString() {
        return "Staff{" +
                " last='" + last + '\'' +
                ", first='" + first + '\'' +
                ", id='" + id + '\'' +
                ", Monthly salary=" + monthlyEarning() + '\'' +
                '}';
    }
}
```

```java
/**
* This class creates a level of education for employees
* each education has a level of Degree, a Major, and amount of research
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
public class Education implements EmployeeInfo {
    String Degree;
    String Major;
    int Research;
    //default constructor leaves variables blank
    public Education(){
        Degree = "NONE";
        Major = "NONE";
        Research = 0;
    }
    //constructor that takes in a degree, major and research
    public Education(String Degree, String Major, int Research){
        this.Degree = Degree;
        this.Major = Major;
        this.Research = Research;
    }
    //setters and getters for degree, major, and research
    public String getDegree(){
        return Degree;
    }
    public void setDegree(String newDegree){
        Degree = newDegree;
    }
    public String getMajor(){
        return Major;
    }
    public void setMajor(String newMajor){
        Major = newMajor;
    }
    public int getResearch(){
        return Research;
    }
    public void setResearch(int newResearch){
        Research = newResearch;
    }

}
```

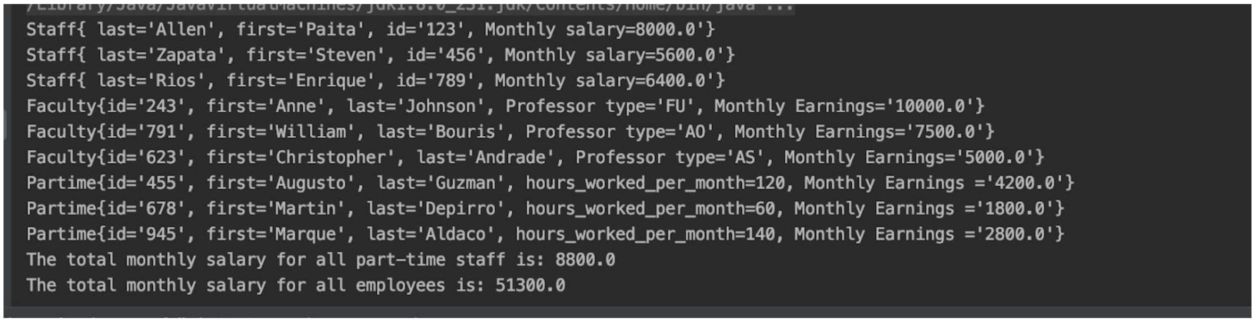
```java
/**
* Faculty is a class that is a sublclass of employee and inherits its variables(first,last, and id)
* Faculty contains an enum which defines what level the faculty is(assistant professor, associate professor, professor)
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
public class Faculty extends Employee {
    //enum with 3 different levels(AS, AO, FU) which represent assistant professor, associate professor, professor
    public enum Level{
        AS, AO, FU
    };
    //creates an instance of level and education
    private Level profs;
    private  Education education;
    //default constructor sets the instance of level we created to as and creates and nee education
    public Faculty(){
        profs = Level.AS;
        education = new Education();
    }
    //constructor that takes in the variables from employee as well as a type of professor(enum) and an
education(object)
    public Faculty(String last, String first, String id, Level profs, Education education){
        super(last, first, id);
        this.profs = profs;
        this.education = education;
    }
    //setters and getters for profs and education
     private Level getProfs(){
        return profs;
     }
     private void setProfs(Level newProfs){
        profs = newProfs;
     }
     private Education getEducation(){
        return education;
     }
     private void setEducation(Education newEducation){
        education = newEducation;
     }

     //overides the monthly earnings in employee and returns the monthly earning bsed of what
    //type of professor they are
    @Override
    public double monthlyEarning() {
        if(profs == Level.AS){
            return FACULTY_MONTHLY_SALARY;
        }
        else if(profs == Level.AO){
            return FACULTY_MONTHLY_SALARY * 1.5;
        }
        else if(profs == Level.FU){
            return FACULTY_MONTHLY_SALARY * 2.0;
        }
        else{
            return 0;
        }
    }

    //to string that prints the variables from employee as well as the professor type and the monthly earning
    @Override
    public String toString() {
        return "Faculty{" +
                "id='" + id + '\'' +
                ", first='" + first + '\'' +
                ", last='" + last + '\'' +
                ", Professor type='" + profs + '\'' +
                ", Monthly Earnings='" + monthlyEarning() + '\'' +
                '}';
    }
}
```

```
/**
* Partime is a class that extends and adds a variable that keeps track of the hours worked per week
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
public class Partime extends Staff implements EmployeeInfo {//implements the interface employeeInfo
    int hours_worked_per_week;
    //default constructor taht sets hours worker per week to 0
    public Partime(){
        super();
        hours_worked_per_week = 0;
    }
    //constructor that takes in the variables from staff as well as hours worked per week
    public Partime(String last, String first, String id, double hourly_rate, int hours_worked_per_week) {
        super(last, first, id, hourly_rate);
        this.last = last;
        this.first = first;
        this.id = id;
        this.hourly_rate = hourly_rate;
        this.hours_worked_per_week = hours_worked_per_week;
    }
    //setter and getter for hours worked per week
    private int getHours_worked_per_week(){
        return hours_worked_per_week;
    }
    private void setHours_worked_per_week(int newHours){
        hours_worked_per_week = newHours;
    }

    //returns the monthly earning which is based on the hourly rate and the hours_worked_per_week
    @Override
    public double monthlyEarning() {
        return hourly_rate * (hours_worked_per_week * 4);
```

```java
    }

    //to string that prints the variables from staff as well as the hours worked per month
    @Override
    public String toString() {
        return "Partime{" +
                "id='" + id + '\'' +
                ", first='" + first + '\'' +
                ", last='" + last + '\'' +
                ", hours_worked_per_month=" + (hours_worked_per_week * 4) +
                ", Monthly Earnings ='" + monthlyEarning() + '\'' +
                '}';
    }
}
```

```java
/**
* public interface that creates two constants which
* will be used in other classes that implement EmployeeInfo
* 1)FACULTY_MONTHLY_SALARY
* 2)STAFF_MONTHLY_HOURS_WORKED
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
public interface EmployeeInfo {
    double FACULTY_MONTHLY_SALARY = 5000.00;
    int STAFF_MONTHLY_HOURS_WORKED = 160;
}
```

```java
import java.util.*;
/**
* This is the tester class that creates 9 instances of employees
* -3 are staff
* -3 are faculty
* -3 are partime workers
* in the tester we add the 9 instances employees into an araylist and print all their toStrings and monthly earnings
* @author Liam Abalos
* @author Mark Fastner
* CECS 277 Lab (Lab #3)
*/
import java.util.ArrayList;

public class Tester{
    public static void main(String[] args){
        double total_monthly_salary_partime = 0;
        double total_monthly_salary_employees = 0;
        //creates 9 objects
        Staff s1 = new Staff("Allen", "Paita", "123", 50.00);
        Staff s2 = new Staff("Zapata", "Steven", "456", 35.00);
        Staff s3 = new Staff("Rios", "Enrique", "789", 40.00);

        Education e1 = new Education("Ph.D", "Engineering", 3);
        Faculty f1 = new Faculty("Johnson", "Anne", "243", Faculty.Level.FU, e1);
        Education e2 = new Education("Ph.D", "English", 1);
        Faculty f2 = new Faculty("Bouris", "William", "791", Faculty.Level.AO, e1);
        Education e3 = new Education("MS", "Physical Education", 0);
        Faculty f3 = new Faculty("Andrade", "Christopher", "623", Faculty.Level.AS, e1);

        Partime p1 = new Partime("Guzman", "Augusto", "455", 35.00, 30);
        Partime p2 = new Partime("Depirro", "Martin", "678", 30.00, 15);
        Partime p3 = new Partime("Aldaco", "Marque", "945", 20.00, 35);

        //creates arraylist and adds object to list
        ArrayList<Employee> employees = new ArrayList<Employee>(9);
        employees.add(s1);
        employees.add(s2);
        employees.add(s3);
        employees.add(f1);
        employees.add(f2);
        employees.add(f3);
        employees.add(p1);
        employees.add(p2);
        employees.add(p3);

        //goes through arraylist and prints out hte toString as well as monthly earnings of each employee
        for(Employee temp: employees){
            total_monthly_salary_employees += temp.monthlyEarning();
            if(temp instanceof Partime){
                total_monthly_salary_partime += temp.monthlyEarning();
            }
            System.out.println(temp);

        }
        System.out.println("The total monthly salary for all part-time staff is: " + total_monthly_salary_partime);
        System.out.println("The total monthly salary for all employees is: " + total_monthly_salary_employees);




    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk1.8.0_231.jdk/Contents/Home/bin/java ...
Staff{ last='Allen', first='Paita', id='123', Monthly salary=8000.0'}
Staff{ last='Zapata', first='Steven', id='456', Monthly salary=5600.0'}
Staff{ last='Rios', first='Enrique', id='789', Monthly salary=6400.0'}
Faculty{id='243', first='Anne', last='Johnson', Professor type='FU', Monthly Earnings='10000.0'}
Faculty{id='791', first='William', last='Bouris', Professor type='AO', Monthly Earnings='7500.0'}
Faculty{id='623', first='Christopher', last='Andrade', Professor type='AS', Monthly Earnings='5000.0'}
Partime{id='455', first='Augusto', last='Guzman', hours_worked_per_month=120, Monthly Earnings ='4200.0'}
Partime{id='678', first='Martin', last='Depirro', hours_worked_per_month=60, Monthly Earnings ='1800.0'}
Partime{id='945', first='Marque', last='Aldaco', hours_worked_per_month=140, Monthly Earnings ='2800.0'}
The total monthly salary for all part-time staff is: 8800.0
The total monthly salary for all employees is: 51300.0
```