**Credit Name:** Inheritance and Polymorphism
**Assignment Name:** UEmployee, Faculty, Staff

How has your program changed from planning to coding to now? Please explain?

```
private String fName, lName;
private double salary;

//Accessor methods
public String getFName()
{
    return(fName);
}

public String getLName()
{
    return(lName);
}

public double getSalary()
{
    return(salary);
}
```

I started by creating an abstract class for UEmployee and added the required member variables and accessor methods for those variables.

```
public String toString()
{
    return("Employee" + fName + " " + lName + "'s salary is: " + salary);
}
```

I then added a toString method and started work on the constructor.

```
//Constructor
public UEmployee(String fN, String lN, double s)
{
    fName = fN;
    lName = lN;
    salary = s;
}
```

For the constructor I just added code that allows you to set the employee's name and salary.

```java
//Modifier methods
public void setFName(String f)
{
    fName = f;
}

public void setLName(String l)
{
    lName = l;
}

public void setSalary(double s)
{
    salary = s;
}
```

I also added the modifier methods and moved on to the faculty class.

```java
public String toString()
{
    return "The department that this faculty member is in is: " + department;
}
```

Finally I added a toString method to return the object as a string.

```java
public class Staff extends UEmployee
{
    private String jobTitle;


    //Constructor
    public Staff(String fN, String lN, double s, String j)
    {
        super(fN, lN, s);
        jobTitle = j;
    }



    //Accessor method
    public String getJobTitle()
    {
        return jobTitle;
    }

    //Modifier method
    public void setJobTitle(String d)
    {
        jobTitle = d;
    }



    //Return object as string
    public String toString()
    {
        return super.toString() + " and the job they have is: " + jobTitle;
    }
}
```

I reused the code from faculty for staff and changed department to jobTitle.

```java
Staff staffMem = new Staff("Bob", "Kid", 60000, "Sec");
Faculty facultyMem = new Faculty("Jack", "brown", 100000, "Professer");
double choice;
int empNum;
```

I then started working on the test code for university by creating objects for 2 employees and declaring variables for choices that would be made later on.

```
//Get an employee number from the user to access/edit
Scanner input = new Scanner(System.in);
System.out.println("Please enter the employee's number: ");
empNum = input.nextInt();
```

I added code to ask the user what employee number they have and then record that input.

```
//Check employee number and display their name
switch(empNum)
{
case 1:
    System.out.println("You are now accessing " + staffMem.getFName() + " " + staffMem.getLName() + "'s data.\n");
break;
case 2:
    System.out.println("You are now accessing " + facultyMem.getFName() + " " + facultyMem.getLName() + "'s data.\n");
break;
}
```

Next I added a welcome message to tell the user the name of the employee that they are accessing.

```
//Loop menu and actions associated with the menu
do
{
    System.out.println("What would you like to do?"
            + "1. View employee's salary."
            + "2. View employee's job/department."
            + "3. Edit employee's name."
            + "4. Edit employee's salary."
            + "5. Edit employee's job/department."
            + "6. See general overview."
            + "7. Quit.\n");

    choice = input.nextInt();



} while(choice != 7);
```

Then I added a loop that would have a menu, and perform actions associated with that menu later. I also add code to quit the program when the user chooses to.

```
//Check employee number and display their name
switch(empNum)
{
case 1:
    emp = staffMem;
break;
case 2:
    emp = facultyMem;
break;
default:
    System.out.println("Defaulting to employee 1.\n");
    emp = staffMem;
}
System.out.println("You are now accessing " + emp.getFName() + " " + emp.getLName() + "'s data.\n");
```

I went back to edit the original code for the intro message for the user to add a UEmployee object that would copy the data of one of the other objects. This would help reduce the switch statements needed later.

```
//Display salary
case 1:
    System.out.println("Their salary is: " + money.format(emp.getSalary()) + "\n");
    break;
```

I started out with an easier one which just prints the salary. I had to add the money format as well.

```
//Display job title/department
case 2:
    if (empNum == 1)
    {
        System.out.println("The job they have is: " + staffMem.getJobTitle() + ".\n");
    }
    else
    {
        System.out.println("The department they're in is: " + facultyMem.getDepartment() + ".\n");
    }
    break;
```

I then created the case for when the user wants to display the job title or department. I needed to add an if statement to check which employee they chose.

```
//Edit name
case 3:
    System.out.println("Would you like to change the first name(1) or last name(2)?");
    int nameType = input.nextInt();

    System.out.println("What is the new name?");
    input.nextLine();
    String name = input.nextLine();



    if (nameType == 1 && empNum == 1)
    {
        staffMem.setFName(name);
    }
    else if (nameType == 2 && empNum == 1)
    {
        staffMem.setLName(name);
    }
    else if (nameType == 1 && empNum == 2)
    {
        facultyMem.setFName(name);
    }
    else
    {
        facultyMem.setLName(name);
    }
    break;
```

Then I added the code for changing the first name and last name, using if else statements to check which name they wanted to change and which employee.

```
//Edit salary
case 4:
    System.out.println("What is the new salary?");
    double salary = input.nextDouble();


    if (empNum == 1)
    {
        staffMem.setSalary(salary);
        emp = staffMem;
    }
    else
    {
        facultyMem.setSalary(salary);
        emp = facultyMem;
    }
    break;
```

Next I added the code to edit the salary, following the same format of checking which employee was picked and using the corresponding objects. I also did the same thing for editing the job title and department name.

```
//Display general info
case 6:
    System.out.println(emp + ".\n");
    break;




//End message
case 7:
    System.out.println("This program will now terminate.");
    break;
```

The last two cases just display the toString method and tell the user that the program ended.

```
/*
 Program: University.java          Date: November 4, 2024
 Purpose: To test the UEmployee, Staff, and Faculty classes.
 Author: Logan Yuen
 School: CHHS
 Course: Computer Science 30
 */
```

Finally I added headers and a screendump.