CSE 460: Software Analysis and Design

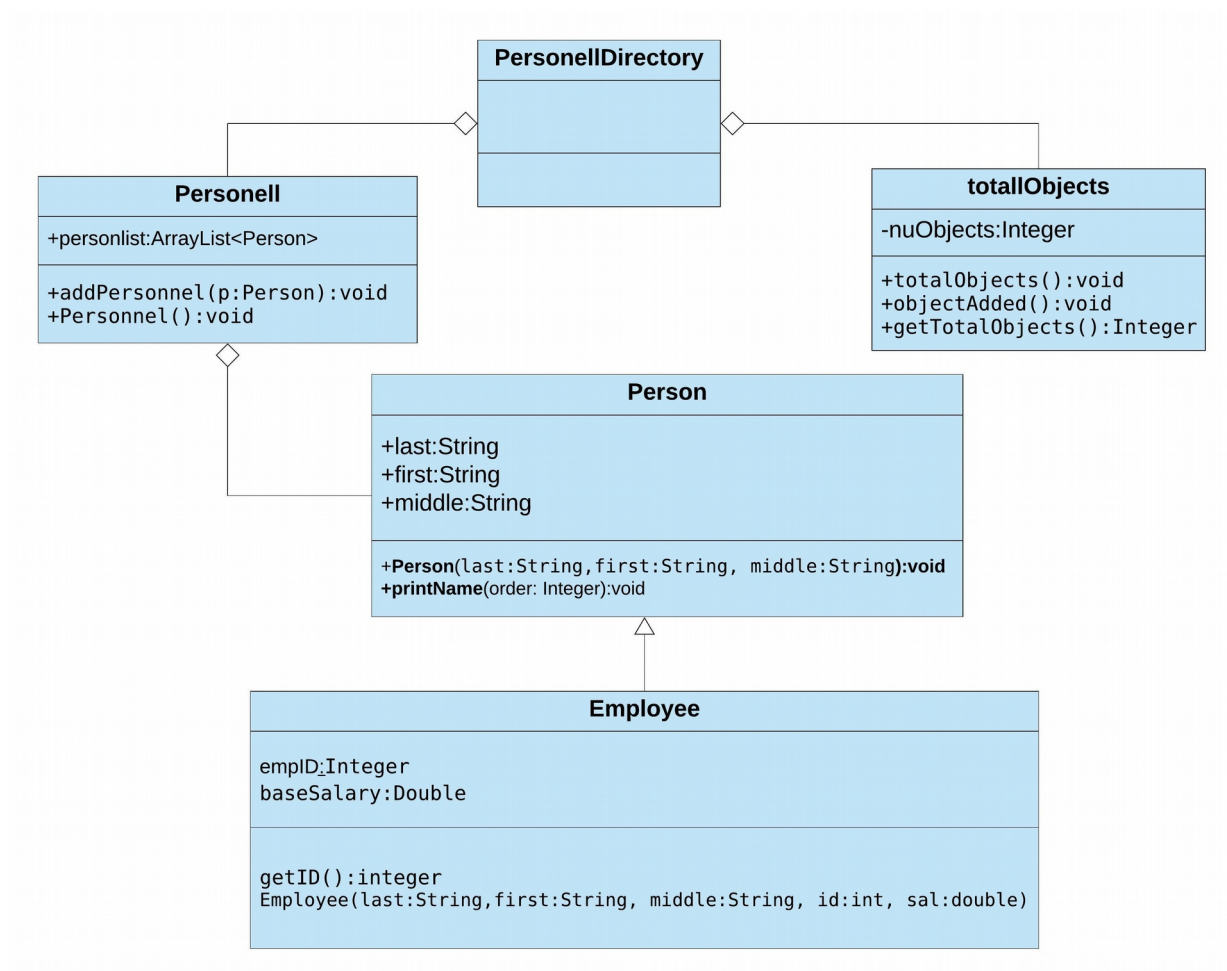**Directory Management System Phase I Submission**

Directions:

Complete your work for Phase I Parts 1 and 2 in this document. Save and submit as a **single** PDF titled "Last Name_First Name_Directory Management System Project_Phase I_Submission".

Phase I, Part 1

Use the Astah tool to draw the class diagram for the current implementation of the university system. Use correct UML notations. When you have completed the diagram, take a clear screenshot and paste it in the space provided.

PersonellDirectory

Personell

+personlist:ArrayList<Person>

+addPersonnel(p:Person):void
+Personnel():void

totallObjects

-nuObjects:Integer

+totalObjects():void
+objectAdded():void
+getTotalObjects():Integer

Person

+last:String
+first:String
+middle:String

+**Person**(last:String,first:String, middle:String**):void**
+**printName**(order: Integer):void

Employee

empID:Integer
baseSalary:Double

getID():integer
Employee(last:String,first:String, middle:String, id:int, sal:double)

Phase I, Part 2

In the code, identify object-oriented concept violations, content coupling, common coupling, control coupling and stamp coupling situations. Copy and paste the code segments that shows each coupling situation in the space provided. *You may use additional space as necessary.*

1. **Object-Oriented Concept Violations**
    content coupling :
    ```
    per.personList.add(p1);
    ```

    Common Coupling
    ```
        if( per.personList.get(i).first.equals(firstN) &&
    per.personList.get(i).last.equals(lastN))
    ```

    Control Coupling

    ```
    per.personList.get(loc).printName(0);

    public void printName(int order)
        {

          if(order == 0)
          {
             System.out.println(first + "  " + middle + "  " + last);

          }else if(order == 1)
             {

             System.out.println(last + " ," + middle + " " + first);

             }
             else if(order == 2)
                       {

                       System.out.println(last + " ," + first + " " +
    middle);

             }
        }
    ```

    Stamp Coupling
    ```
     for(int i=0; i<per.personList.size(); i++)
       {
        per.personList.get(i).printName(order);
       }
    ```

How would you fix these violations?

Content coupling: I would use the already inbuilt method for adding personnel to add a new personell

Common Coupling : use getters and setters to access the values of **Person**
   Control Coupling   : create other methos that can be called to print a specific option
Stamp Coupling   :

2. **Content Coupling**

```
per.personList.add(p1);
```

How would you fix this?

I would use the method `addPersonnel(Person p) to access   the personList`

3. **Common Coupling**

```
if( per.personList.get(i).first.equals(firstN) &&
per.personList.get(i).last.equals(lastN))
```

How would you fix this?

I would make the variables first and last in person private and use getters and setters to obtain their values. That is using data hiding

4. **Control Coupling**
```
per.personList.get(loc).printName(0);

public void printName(int order)
   {

      if(order == 0)
      {
         System.out.println(first + "  " + middle + "  " + last);

      }else if(order == 1)
         {

         System.out.println(last + " ," + middle + " " + first);

         }
         else if(order == 2)
                  {
```

```
                    System.out.println(last + " ," + first + " " +
middle);

        }
}
```

How would you fix this?

I would create methods that would do the same thing differently, one for each of the
situations

5. **Stamp Coupling**

```
for(int i=0; i<per.personList.size(); i++)
{
 per.personList.get(i).printName(order);
}
```