**Human Resources Management System**
By Sunhwa Lee

**Documentation**

    I.    MainFrame
        a.  addTeacherSubmitActionPerformed()

```
try
      ask user for inputs needed to create teacher object
      throw exception if any field is left blank

      if full time employee
            hoursWorked = 32
      else
            use hours specified

      create teacher object given all parameters

      if teacher already exists in department
            throw exception teacher exists
      else
            add teacher object to department
            call saveProgress method to write to text
file
            update dashboard display with new object
            clear form fields

catch exception
      show dialog stating invalid submission
```

        b.  staffSubmitActionPerformed()

```
exactly same as pseudocode above (I.a.) but applied for
staff
```

    II.    Driver
        a.  initialize()

```
create initial ArrayList of class Department
create sample deans of class Teacher
create three Departments with respective deans
store to ArrayList
write to file
```

        b.  saveProgress(ArrayList<Department> departments)

```
try
```

```
        create file output stream
        create object output stream
        for each department in ArrayList departments
                write each object to file
        close output streams

catch exception
        print exception
```

c. readProgress()

```
create ArrayList of department
try
        create file input stream
        create object input stream
        while file has content available
                read objects
                add to ArrayList
        close input streams

catch exception
        print exception
```

d. newTeacher()

```
create scanner object
ask user for all fields needed to create teacher object
return new object of Teacher given fields
```

e. newStaff()

```
create scanner object
ask user for all fields needed to create staff object
return new object of staff given fields
```

III. Department
    a. display()

```
create String variable display
append dean information
append all teachers information
append all staff information
return display
```

b. addTeacher(Teacher teacher)

```
add teacher to ArrayList of class Teacher
```

c. addStaff(Staff staff)

```
add staff to ArrayList of class Staff
```

IV. Person

a. display()

```
create String variable display
append all data members of person
return display
```

V. Teacher

a. getDegreeRate()

```
return switch case using degree variable
      case 1 is 42
      case 2 is 82
      case 3 is 112
      default case is 0
```

b. defineCategory()

```
return  "Full-Time Teacher if fullTime is true else
"Part-Time Teacher"
```

c. ComputePayRoll()

```
if fullTime is true
      return (32 * degreeRate() * 2) * 0.85
else
      return (hoursWorked * degreeRate() * 2) * 0.76
```

d. display()

```
create String variable display
append all details about teacher
compute and append PayRoll
return display
```

VI. Staff

a. defineCategory()

```
return "Staff"
```

b. ComputePayRoll()

```
return (workload * 32 * 2) * 0.75
```

c. display()

```
create String variable display
append all details about Staff
return display
```