Min Kyaw
018182136

Code edited in EmployeeFactory.java

```
//Create other cases (Staff and Parttime)

        else if (employeeType.equalsIgnoreCase("STAFF")) {
            return new Staff();
        }

        else if (employeeType.equalsIgnoreCase("PARTIME")) {
            return new Partime();
        }

        return null;
    }
}
```

Code edited in Faculty.java

```
public void setEmployee()
    {
        //Input last name, first name, id, gender, birthday, Faculty level (Assistant professor<
        //Associate professor, or Full Professor), degree, major and research.
        //YOUR CODE

        Scanner in = new Scanner(System.in);
        System.out.println("Enter Last Name: ");
        this.setLastName(in.next());
        System.out.println("Enter First Name: ");
        this.setFirstName(in.next());
        System.out.println("Enter ID Number: ");
        this.setID(in.next());
        System.out.println("Enter Gender: ");
        this.setGender(in.next().charAt(0));
        System.out.println("Enter the birthday in the following order: MM/DD/YYYY ");
        int m = in.nextInt();
        int d = in.nextInt();
        int y = in.nextInt();
        this.setBirthday(m, d, y);
        System.out.println("Enter this faculty employee's level from three choices: ");
        System.out.println("1) Assistant Professor: ");
        System.out.println("2) Associate Professor:");
        System.out.println("3) Full Time Professor: ");
        System.out.println("Choice: ");
        int LvlInput = in.nextInt();

        if (LvlInput == 1) {
           this.setRank(Level.ASSISTANT_PROFESSOR);
        }

        else if (LvlInput == 2){
           this.setRank(Level.ASSOCIATE_PROFESSOR);
        }

        else {
           this.setRank(Level.FULL);
        }

        changeEdu();

    }
```

Code edited in Staff.java

```
public void setEmployee()
    {
        // Input last name, first name, id, Gender, birthday, and hourly rate
        Scanner in = new Scanner(System.in);
        System.out.println("Enter Last Name: ");
        this.setLastName(in.next());
        System.out.println("Enter First Name: ");
        this.setFirstName(in.next());
        System.out.println("Enter ID Number: ");
        this.setID(in.next());
        System.out.println("Enter Gender: ");
        this.setGender(in.next().charAt(0));
        System.out.println("Enter the birthday in the following order: MM/DD/YYYY ");
        int m = in.nextInt();
        int d = in.nextInt();
        int y = in.nextInt();
        this.setBirthday(m, d, y);
        System.out.println("Enter the hourly rate: ");
        double HR = in.nextDouble();
        this.setHourlyRate(HR);
    }
}
```

Code edited in Partime.java

```
public void setEmployee()
   {
      //Input last name, first name, id, Gender, birthday, hourly rate and hours work
      Scanner in = new Scanner(System.in);
      System.out.println("Enter Last Name: ");
      this.setLastName(in.next());
      System.out.println("Enter First Name: ");
      this.setFirstName(in.next());
      System.out.println("Enter ID Number: ");
      this.setID(in.next());
      System.out.println("Enter Gender: ");
      this.setGender(in.next().charAt(0));
      System.out.println("Enter the birthday in the following order: MM/DD/YYYY ");
      int m = in.nextInt();
      int d = in.nextInt();
      int y = in.nextInt();
      this.setBirthday(m, d, y);
      System.out.println("Enter the hourly rate: ");
      double HR = in.nextDouble();
      this.setHourlyRate(HR);
      System.out.println("Enter the hours worked: ");
      double HW = in.nextDouble();
      this.setHoursWorked(HW);

   }
}
```

Code edited in Tester_Employee.java

```
//Create an object of EmployeeFactory
      EmployeeFactory empf = new EmployeeFactory();
      // ____________________________________________

      employeeList = new ArrayList<Employee>();

      //Save a binary file that contains a collection of employees

      File f = new File("Employees.dat");

      if(f.exists())
      {
          ObjectInputStream input = new ObjectInputStream(new FileInputStream(f));
          employeeList = (List<Employee>) input.readObject();
          input.close();
      }

      else
      {
          employeeList = new ArrayList<Employee>();
      }

      int menuchoice = 0;
      while (menuchoice != 5)
      {
          System.out.println("\nEmployee Information\n");
          System.out.println("1) Add an employee");
          System.out.println("2) Display employees in sorted orders");
          // System.out.println("3) Total monthly salary");
          // System.out.println("4) Test Cloneable");
          System.out.println("5) Quit");
          System.out.println("Enter choice:");
          menuchoice = in.nextInt();
          switch (menuchoice)
          {
              case 1:
                  System.out.println("What type of employee would you like to add?");
                  System.out.println("Enter 'Staff':");
                  System.out.println("Enter 'Faculty':");
                  System.out.println("Enter 'Partime':");
                  System.out.println("Enter choice:");
                  String choice = "";
                  choice = in.next();
//                      if (choice.equalsIgnoreCase("FACULTY")) {
//                          empf.getEmployee(choice);
//
//                      }
//                      Employee emp1 = empf.getEmployee("STAFF");
//                      Employee emp2 = empf.getEmployee("FACULTY");
//                      Employee emp3 = empf.getEmployee("PARTIME");

                  //Your code to call the getEmployee method from the class EmployeeFactory to
input an employee

                  //Call the setEmployee method
                  if (choice.equalsIgnoreCase("STAFF")) {
                      emp = empf.getEmployee(choice);
                      emp.setEmployee();
                  }

                  else if (choice.equalsIgnoreCase("FACULTY")) {
                      emp = empf.getEmployee(choice);
                      emp.setEmployee();
                  }

                  else if (choice.equalsIgnoreCase("PARTIME")) {
                      emp = empf.getEmployee(choice);
                      emp.setEmployee();
                  }

                  //Add employee to the list

                  employeeList.add(emp);

                  break;
              case 2:
                  System.out.println("1) Display employees sorted by ID number in ascending
order");
                  System.out.println("2) Display employees sorted by last name in descending order
\n");
                  int opt = 0;
                  opt = in.nextInt();
                  switch(opt)
                  {
                      case 1:
                          System.out.println("Part D: Sorting by ID number in ascending order...");
                          Collections.sort(employeeList, new EmployeeIdComparator());
                          if(employeeList.isEmpty())
                          {
                              System.out.println("There are no employees!\n");
                          }
                          else
                          {
                              for(Employee e: employeeList)
                              {
                                  System.out.println(e.toString());
                              }
                          }
                          break;
                      case 2:
                          System.out.println("Part E: Sorting by Last Name in descending order...");
                          Collections.sort(employeeList);
                          if(employeeList.isEmpty())
                          {
                              System.out.println("There are no employees!\n");
                          }
                          else
                          {
                              for(Employee e: employeeList)
                              {
                                  System.out.println(e.toString());
                              }
                          }
                          break;
                      default:
                          System.out.println("Incorrect choice");
                          break;
                  }
                  break;
//                case 3:
//                    System.out.println("1) Display Total Monthly Salary for Part Time Staff \n2) Total
Monthly Salary for All Employees");
//                    int option = 0;
//                    option = in.nextInt();
//                    switch(option)
//                    {
//                        case 1:
//                            System.out.println("Part B: Total Monthly Salary for Part Time Staff");
//                            double total_part_time = 0.0;
//                            for (Employee e1: employeeList)
//                            {
//                                if(e1 instanceof Partime)
//                                {
//                                    total_part_time += e1.monthlyEarning();
//                                }
//                            }
//                            System.out.println("  " + "$" + total_part_time + "\n");
//                            break;
//                        case 2:
//                            System.out.println("Part C: Total Monthly Salary for All Employees");
//                            double total = 0.0;
//                            for (Employee e2: employeeList)
//                            {
//                                total += e2.monthlyEarning();
//                            }
//                            System.out.println("  " + "$" + total + "\n");
//                            break;
//                        default:
//                            System.out.println("Incorrect choice!");
//                            break;
//                    }
//                    break;
//                case 4:
//                    System.out.println("Part F: Test Cloneable...");
//                    Employee original = null;
//                    original = employeeFactory.getEmployee("FACULTY");
//                    original.setEmployee();
//                    Faculty cloned = (Faculty) original.clone();
//                    System.out.println("Original:");
//                    System.out.println(original);
//                    System.out.println("Cloned:");
//                    System.out.println(cloned);
//                    System.out.println("Change education to prove deep copy:");
//                    cloned.changeEdu();
//                    System.out.println("Original:");
//                    System.out.println(original);
//                    System.out.println("Cloned:");
//                    System.out.println(cloned);
//                    break;
               case 5:
                    System.out.println("Goodbye!");
                    break;

                default:
                    System.out.println("Incorrect choice \nTry again!\n");
            }
        }

        ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(f));
        out.writeObject(employeeList);
        out.close();
    }
}
```