Lab 1 Solutions

1. Instance variables in class `Person` are public and are used by the main program (see class `Main1`) resulting in ***content coupling***.

   Implemented solution to remove ***content coupling*** is a Java project that is stored under Solutions/Lab1-v1. You can simply import Lab1-v1 under the folder named Solutions into your Eclipse workspace.

   As you will se in the implemented solution, to remove ***content coupling***, all instance variables in class `Person` have been made **private** and the following methods were implemented in class `Person` to be used by the main program of class `Main1:`

```
public void setType(int person_type) {
        this.type = person_type;
}

public void initWorker() {
        this.worker = new Person[max_nworkers];
}

public void setBoss(Person person_boss) {
        this.boss = person_boss;
}

public void addWorker(Person person_worker) {
        this.worker[nworkers++] = person_worker;
}
```

2. The constructor in class `Person` has more than three parameters (it specifically has four parameters), resulting in ***data coupling.*** Implemented solution to remove ***data coupling*** is a Java project that is a modified version of Lab1-v1 and is stored under Solutions/Lab1-v2. You can simply import Lab1-v2 under the named folder Solutions into your Eclipse workspace.

   As you will see in the implemented solution, a new class `dob` has been added to the source code and the constructor `Person` is modified as follows:

```
public Person(String n, dob person_birth_date) {
        name = n;
        birth_date = person_birth_date;
                                                把类当作C语言里面的结构体来用
        boss = null;
        worker = null;
        nworkers = 0;
}
```

   Moreover, in class `dob` the following methods have been implemented to be used by the main program of class `Main1:`

```
public int get_dob_day() {
        return this.dob_d;
}
```

```
public int get_dob_month() {
        return this.dob_m;
}

public int get_dob_year() {
        return this.dob_y;
}
```

3. In the source code you have been working on during this lab session (i.e., the Java Project titled Lab1.), creating an instance of class `Person` takes several steps:

- setting the `name` and `dob` in the constructor, 必须得一起执行,忘记执行其中一个就会出现错误
- setting the `type` with a separate statement, and then
- setting the `boss` or initializing the `worker` array.

As a result, ***routine coupling*** is introduced.

Implemented solution to reduce ***routine coupling*** is a Java project that is a modified version of Lab1-v2 and is stored under Solutions/Lab1-v3. You can simply import Lab1-v3 under Solutions into your Eclipse workspace.

As you will see in the implemented solution, the constructor `Person` is modified as follows:

```
public Person(String n, dob person_birth_date, int person_type)    {
       name = n;
       birth_date = person_birth_date;
       type = person_type;
       boss = null;
       nworkers = 0;

       if(type == 1) // 1 = Boss
          worker = new Person[max_nworkers];
       else
          worker = null;
}
```

Since `type` is set and `worker` array is initialised in the constructor, the following methods that were implemented in `Lab1-v2` have been removed.

```
public void setType(int person_type) {
       this.type = person_type;
}


public void initWorker() {
       this.worker = new Person[max_nworkers];
}
```

4. The Boss – Worker relationship is in 2 places, implying ***routine coupling***. Implemented solution to remove ***routine coupling*** is a Java project that is a modified version of Lab1-v3 and is stored under Solutions/Lab1-final. You can simply import Lab1-final under the folder named Solutions into your Eclipse workspace.

In the final version of the source code, the following method is added to class `Person` and the `Main` program uses this method rather than separately calling methods `setBoss()` and `addWorker()`.

```
public void addAndSetWorker(Person person_worker) {
          this.addWorker(person_worker);
          person_worker.setBoss(this);
}
```