Lab 8 COMP1161 – Introduction to Object Oriented Programming

POLYMORPHISM

THIS LAB HAS 4 EXERCISES

The Department of Computing occasionally hires both academic staff and administrative staff. Every academic staff member is a lecturer but some are also professors. Every staff member has an identification number, a last name, a first name, and a basic salary.

Exercise #1

Write the class `Staff` to represent a staff member. Your code shall include:

- Declarations for all attributes that belong in this class. These will be those attributes that all staff members have.
- A constructor that gets (as parameters) a staff member's id, last name, first name, and basic salary (in that order).
- An accessor method named getName that returns the staff member's name in the form last name, first name (e.g. "Brown, John")
- A method named `pay` that returns the staff member's basic pay.
- A `toString method` which returns a string in the following format:

```
Staff Member:  last name, first name id#
Basic Pay:     <basic pay amount>
```

Example:

```
Staff Member:        Brown, John     10036572
Basic Pay:           3000.00
```

Exercise #2

1. Write the class `Admin` as a subclass of `Staff`. Your code should include:
   - Declarations of all attributes that an administrative member of staff has which are not already declared in the parent class.
   - A constructor that accepts the staff member's id, last name, first name, basic pay, and overtime rate as arguments.
   - A method called setOvertime that accepts the number of overtime hours worked by an administrative staff member and sets this attribute on the object.
   - A `toString` method that formats its returned value in the following manner:

```
Staff Member:        Brown, John    10036572
Basic Pay:           3000.00
Overtime:               0.00
Total:               3000.00
```

2. Write the class `Lecturer` to represent a lecturer. This class is also a subclass of the `Staff` class. Your code should include:
   - Declarations of all attributes that a lecturer has which are not already declared in the parent class.
   - A constructor that accepts a staff member's id, last name, first name, and basic pay.
   - A `toString` method that formats its returned value in the following manner:

```
Staff Member:        Brown, John    10036572
Basic Pay:           3000.00
```

3. Write the class `Professor` to represent a professor. Your code should include:
   - Declarations of all attributes that a professor has which are not already declared in the parent class.
   - A constructor that accepts a professor's id, last name, first name, basic pay, and rate to be paid per paper.
   - A method named `setPapers` which accepts the number of papers published by a professor and sets this value on the object.
   - A `toString` method that formats its returned value in the following manner:

```
Staff Member:        Brown, John    10036572
Basic Pay:           3000.00
Allowance:              0.00
Total:               3000.00
```

4. Override the `pay` method in each subclass so that pay is calculated correctly for each type of employee.

Exercise #3

1. Write a driver class with a main method which creates an array with the data from the following table and displays the array contents.

Id #	Name	Category	Basic Salary	Additional Data
1000	Brown, John	Administrative	100,000.00	Overtime @ $20/hr
7000	Smith, James	Lecturer	125,000.00	-
3000	Bean, Jim	Professor	300,000.00	Allowance @50/paper
2000	Jones, Martha	Administrative	75,000.00	Overtime @15/hr

2. Modify your code to use the `setOvertime` method to set overtime hours for the element at position 0 to 20 hours.

3. Modify your code to use the `setPapers` method to set the number of papers for the element at position 2 to 12 papers.

4. Display the contents of the array.

Discussion

1. What type of objects did you declare for your array? Explain why.

2. Would your program have worked if you had created an array of objects of type `Lecturer`.

3. Say whether or not the following statements will compile and if not state what modification would have to be made to correct the error.

```
Staff me = new Lecturer(1099, “James”, “John”, 120000);
me.setPapers(3);
```

4. There was one case in which a method was overridden unnecessarily. Can you identify this case?