### lmu_blakCourse Submission Cover Sheet Module: CC4001 Programming Engineering

### Assignment no: 001

### Weighting: 30% of module mark

### Deadline: Friday 20th of January 2023

**Module Leader: Dr Sandra Fernando Student ID:**

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(Signature:) Date:

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations on*

Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples …..

1. **Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.**
2. **Falsifying data in experimental results.**
3. Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
4. **Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.**
5. Collusion to present joint work as the work solely of one individual.
6. Plagiarism, where the work or ideas of another are presented as the candidate's own.
7. Other conduct calculated to secure an advantage on assessment.
8. Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " e = mc2 (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

1

**Assignment**

This assignment will be marked out of 100 and carries 30% of the overall module weighting. Your .java files and report for this part must be uploaded to Weblearn and submitted by 3pm on Friday 20th January 2023. The assignment must be carried out individually so you must not obtain help from anyone other than the module teaching staff. You must not copy code from any source apart from the module core text and the module Weblearn site. Collusion, plagiarism (unreferenced copying) and other forms of cheating constitute Academic Misconduct, which can lead to failure of the module and suspension from the University.

**Aim**

The aim of this assignment is to create a class to represent a **StaffHire**, together with two subclasses to represent Full-time Staff and a Part-time Staff respectively. You will also need to write a report about your program. You are advised to study the learning aim on reflective practice, which is linked from Weblearn. Your report should be no more than 1200 words in length (excluding the table of contents, class diagram and code listing).

**Deliverables**

When you are ready to submit your solution, upload your **StaffHire**, **FullTimeStaffHire** and **PartTimeStaffHire** files (not any other files from the project) to Weblearn together with your report in Microsoft Word format and, when you are absolutely sure that you have uploaded the correct files, press the submit button.

**The program should consist of the following classes.**

The StaffHire class has eight attributes, which correspond to the Vacancy number, Designation, Job Type (Job Type indicates permanent, contact, temporary job) and the staffName, joiningDate, qualification, appointedBy and joined. The Designation, Job Type staffName, joiningDate, qualification and appointedBy are represented as strings of text , vacancy number as a number and joined as a boolean.

The vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy and joined are initialized in the constructor by being assigned values of the constructor's parameters. Each attribute has a corresponding accessor method(i.e. getter and setter methods for all attributes). A display method will output (suitably annotated) the vacancy number, designation and job type, staff Name, join date, qualification, appointed by, joined.

1. Vacancy Number - a whole number
2. Designation Type -a string characters
3. Job Type - a string characters
4. staffName - a string of characters
5. joiningDate - a string of characters
6. qualification - a string of characters
7. appointedBy - a string of characters
8. joined - either true or false (boolean)

There is a method to amend the joined status.

1. The **FullTimeStaffHire**  class is a subclass of the **StaffHire** class and has 2 attributes:

salary - a double number

weeklyFractionalHours - a whole number

The constructor accepts 10 parameters, which vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy, joined, salary and weeklyFractionalHours. A call is made to the superclass constructor with 8 parameters, vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy, joined. The salary and weeklyFractionalHours are assigned the corresponding parameter values.

Each new attribute has a corresponding accessor method.

A method is required to set the salary as changes to salary inevitably occur. The method accepts a new salary as a parameter and, if the staff has joined yet, the new value is assigned to the salary attribute. If the staff is not appointed, then a suitable message is output to the user indicating that that there is no staff appointed to set the salary.

A method is required to set the weeklyFractionalHours as changes to new working hour. The method accepts a new weeklyFractionalHours as a parameter and assigns the new value to the weeklyFractionalHours attribute.

A method to display the details of the staff is required. It must have the same signature as the display method in the StaffHire class. It will call the method in StaffHire class to display the vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy and joined. If the staff salary is set then the salary, weeklyFractionalHours should also be displayed. Each output must be suitably annotated.

1. The **PartTimeStaffHire** class is also a subclass of the **StaffHire** class and it has 4 attributes:

workingHour - a whole number

wagesPerHour -a double number

shifts - a string of characters

terminated - either true or false (boolean)

The constructor accepts 11 parameters, which are the vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy and joined, working hours (hours per day), wages per hour, and shifts (morning, day or evening working shifts). A call is made to the superclass constructor with 8 parameters, the vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy and joined. The working hours, wages per hour and shifts attributes are assigned the corresponding parameter values. Additionally, in the constructor, assign terminated status is assigned to false.

Each new attribute has a corresponding accessor method.

A method is required to set the working shifts as changes to shifts inevitably occur. The method accepts a new shifts as a parameter and assigns the new value to the shifts attribute if the staff has joined.

There is a method to terminate the staff. If the staff is already terminated, a suitable message is displayed. Else, staff name, join date, qualification, appointed by are set to an empty string. Similarly, the joined status is set to false while the terminated status is changed to true.

A method to display the details of the part time staff is required. It must have the same signature as the display method in the StaffHire class. It will call the method in StaffHire class to display the vacancyNumber, designation, jobType, staffName, joiningDate, qualification, appointedBy and joined. If the working hours are set then wages per hour, working hour, shifts and terminated should also be displayed. Income per day is the product of wages per hour and working hour per day should also be displayed. Each output must be suitably annotated.

1. Additional marks will be awarded for good program style, particularly naming, layout and documentation. See http://www.bluej.org/objects-first/styleguide.html for details.

**Report**

The report should contain:

* A class diagram showing the inheritance relationship between the three classes and, for each class, the name of the class, the data types and names of the fields and the return types and signatures of the methods
* A short description of what each method does
* You should give evidence (through appropriate screenshots) of the following testing that you carried out on your program:
* Test 1: Inspect StaffHire **hire a person**, re-inspect the StaffHire
* Test 2: Inspect FullTimeStaffHire, **add salary**, re-inspect the the FullTimeStaffHire
* Test 3: Call the method to **display** the details of the FullTimeStaffHire class
* Test 4: Inspect PartTimeStaffHire, **add working shifts**, PartTime
* Test 5: StaffHire Inspect PartTimeStaffHire, **terminate a staff**, PartTimeStaffHire
* Test 6: Call the method to **display** the details of PartTimeStaffHire,

The report should contain a section on error detection and error correction where you give examples and evidence of three errors encountered in your implementation. The errors (syntax and/or runtime) should be distinctive and not of the same type.

The report should contain a conclusion, where you evaluate your work, reflecting on what you learnt from the assignment. You are advised to study the learning aid on reflective practice, which is linked from the module in WebLearn.

The report should include a title page, a table of contents (with page numbers), and a listing of the code (in an appendix). Marks will also be awarded for the quality of writing and the presentation of the report.

|  |  |
| --- | --- |
| **A picture containing text  Description automatically generated**  **School of Computing and Digital Media**  **Marking Criteria for Coursework** | |
| **Marking criteria** | **Mark range** |
| An exceptional submission, with extensive and detailed knowledge based on a high level of additional background research; a high degree of critical analysis, evaluation and original insight; excellent organization and presentation. | 90 – 100 |
| In addition to the requirements for grades of 70-79% (below), an outstanding submission incorporating a high level of originality, depth and critical insight and going well beyond expected work. | 80 – 89 |
| An excellent submission, going beyond expected work; commanding understanding and appreciation of the central points; well-written and effectively structured; evidence-based, critical and logical analysis; comprehensive and correct referencing of sources. | 70 – 79 |
| A very good and comprehensive submission that fulfills the assignment brief; relevance and accuracy; clear structure and evidence-based; a sound grasp of the subject and ability to think about it effectively and critically; correct referencing of sources. | 60 – 69 |
| A solid submission that fulfills most of the assignment brief; adequate structure; mostly accurate, with few errors or omissions; some limitations in scope, critical thinking and argument; a consistent attempt at referencing sources. | 50 – 59 |
| A basic but incomplete submission, with limited relevant information; lacks logical and coherent structure, with some significant errors or omissions; contains sparse and/or irrelevant information and lacks an evidence-based approach; some limitations with referencing of sources; poor grammar and spelling. | 40 – 49 |
| An unsatisfactory submission, with some substantial errors, omissions or irrelevancies; barely acceptable amount of information relevant to the question; poor structure, presentation and expression and referencing of sources. | 30 – 39 |
| An unsatisfactory submission, that doesn’t meet the learning outcomes with many substantial errors, misconceptions, omissions or irrelevancies; little information relevant to the question; very poor structure, presentation and expression. | 1 – 29 |
| Non-submission or submission of work which cannot be given any credit (e.g., blank submission, incorrect assignment). | 0 |