1st SIT COURSEWORK QUESTION PAPER: **Year Long Spring 2019**

Module Code:	**CS4001NA**
Module Title:	**Programming**
Module Leader:	**Chirag Thapa** (Itahari International College)

Coursework Type:	**Individual** **Teacher hiring system**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**12th Week (03:00 PM)**
When Coursework is given out:	**8th Week**
Submission Instructions:	Submit the following to Itahari International College RTE department before the due date: • **A report in PDF format and zip file which includes program file** • **File should be in .java format**
Warning:	London Metropolitan University and Itahari International College takes Plagiarism seriously. Offenders will be dealt with sternly.

Plagiarism Notice

You are reminded that there exist regulations concerning plagiarism.

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 …..

(i) Cheating: including copying coursework.
(ii) Falsifying data in experimental results.
(iii) 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.
(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate’s assessment.
(v) Collusion to present joint work as the work solely of one individual.
(vi) Plagiarism, where the work or ideas of another are presented as the candidate’s own.
(vii) Other conduct calculated to secure an advantage on assessment.
(viii) Assisting in any of the above.

Some notes on what this means for students:

(i) 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.
(ii) 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 reference 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.

Further information in relation to the existing London Metropolitan University regulations concerning plagiarism can be obtained from http://www.londonmet.ac.uk/academic-regulations

Assessment

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 and submitted by 5pm on Friday of Week 12th. 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 materials. Collusion, plagiarism (unreferenced copying) and other forms of cheating constitute Academic Misconduct, which can lead to failure of the module and suspension.

Aim

The aim of this assignment is to create a class to represent a TEACHERS, together with two subclasses to represent a Lecturer and a Tutor respectively. The class that represents a TEACHERS will be based on, but significantly different from, the one that was developed for CC1006NA last semester. You will also need to write a report about your program.

Deliverables

Create a new project in BlueJ and create three new classes (**Teacher**, **Lecturer** and **Tutor**) within the project. Lecturer and Tutor are subclasses of Teacher. When you are ready to submit your solution, upload your ***Teacher.java***, ***Lecturer.java*** and ***Tutor.java*** files (not any other files from the project) together with your report in Microsoft Word format.

Program (60 marks)

The program should consist of the following classes (with no additional attributes or methods).

1) The Teacher class has four attributes, which correspond to the subject, teacher name, interviewer name and classes per day. The subject, teacher name and interviewer name are each represented as a string of text and class per day as a number. The subject, interviewer name and class per day are initialized in the constructor by being assigned the value of the constructor's parameters. The teacher name is assigned an empty string ("").

   Each attribute has a corresponding accessor method. Define a method to set the teacher name to a new name. A display method will output (suitably annotated) the subject, class per day, class name and, if the teacher name is not an empty string, the teacher name too.

[10 marks]

2) The Lecturer class is a subclass of the Teacher class and has seven attributes:

salary	- a whole number
startingFrom	- a string of characters
academicBlockNumber	- a string of characters
dailyWorkingHour	- a number
advanceSalary	- a whole number
joined	- either true or false (boolean)
terminated	- either true or false (boolean)

   The constructor accepts five parameters, which are the subject, class name, class per day, salary and daily working hours. A call is made to the superclass constructor with three parameters, the subject, interviewer name and class per day. The salary and daily working hours attributes are assigned the corresponding parameter values. Additionally, in the constructor, assign starting from as an empty ("") string, academic block number as an empty ("") string, advance salary to 0.0 and the lecturer joined and terminated status is initialized to false.

Each 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 assigns the new value to the salary attribute.

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

There is a method to appoint lecturer for the particular subject. The method accepts, as parameters, a new lecturer name, starting from, advance salary, academic block number – where lecture working station will be. If the lecturer is already joined, an appropriate message including the lecturer name and academic block number is output to the student. If the lecturer has not joined yet, the method to set the lecturer name(teacher name in parent class) is called from parent class with the lecturer name as a parameter. The starting from, academic block number and advance salary are required to update by the parameter values input to the method, the joined status of the lecturer is changed to true and lecturer termination status is initialized to false.

There should be a method for lecture termination. If the lecture is terminated, a suitable message is output. If the lecture is not terminated, the method to set the lecturer name(method to set teacher name in parent class) is called with "" as a parameter, the starting from is set to "", advance salary is set to 0.0, the joined status is set to false and the terminated status is set to true.

There is a method to print the subject, lecturer name and salary, each suitably annotated. (To access the subject and lecturer name, the method to get the subject and teacher name must be called.)

A method to display the details of the lecturer is required. It must have the same signature as the display method in the teacher class. It will call the method in teacher class to display the subject, interviewer name and class per day. If the lecturer is already appointed then termination status, starting from, advance salary and teacher name must also be displayed. Each output must be suitably annotated.

[20 marks]

3) The Tutor class is also a subclass of the Teacher class and it has seven attributes:

salary - a whole number

appointedDate - a string of character

evaluationPeriod - a string of character

terminationDate - a string of character

qualification - a string of character

appointedBy - a string of character

joined - either true or false (Boolean)

The constructor accepts six parameters which are the subject, interviewer name, class per day, salary, appointed by and termination date. A call is made to the superclass constructor with three parameters, the subject, interviewer name and class per day. The attributes salary, appointed by and termination date are given the corresponding parameter values. Additionally, the constructor initializes appointed date to empty ("") string, evaluation period to empty ("") string, qualification to empty ("") string and joined status to false.

Each 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 tutor has not joined yet, the new value is assigned to the salary attribute. If the tutor is already appointed, then a suitable message is output to the user indicating that it is therefore not possible to change the salary.

There is a method for appoint tutor for the particular platform. The method has four parameters, the tutor name, appointed date, termination date, qualification. If the tutor has not joined yet, the method to set the tutor name is called with the tutor name as a parameter and appointed status is set to true, otherwise a suitable message with appointed date is output to the user indicating that the tutor is already appointed. Remaining parameters are assigned to corresponding attributes.

A method to display the details of the tutor is required. It must have the same signature as the display method in the teacher class. It will call the method in teacher class to display the class, subject name and class per day. If the tutor is already joined then appointed date, tutor name, evaluation period, termination date, tutor salary, tutor qualification and tutor appointed by must also be displayed. Each output must be suitably annotated.

[20 marks]

Additional marks will be awarded for good program style, particularly naming, layout and comments. See http://www.bluej.org/objects-

first/styleguide.html for details.

[10 marks]

Report (40 marks)

Your report should describe the process of development of your classes with:

A class diagram
[5 marks]

Pseudocode for each method in each class
[10 marks]

A short description of what each method does
[5 marks]

You should give evidence (through appropriate screenshots) of the following testing that you carried out on your program:

Test 1: Inspect Lecturer Class, appoint lecturer for any particular subject, and re-inspect the Lecturer Class
[2 marks]
Test 2: Inspect Lecturer, change the status of lecturer to terminated, and re-inspect the Lecturer Class
[2 marks]
Test 3: Inspect Tutor class, appoint tutor, and re-inspect.
[2 marks]
Test 4: Display the detail of Lecturer and Tutor Class.
[4 marks]

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. **[3 marks]**

The report should contain a conclusion, where you evaluate your work, reflecting on what you learnt from the assignment, what difficulties you encountered and how you overcame the difficulties.
[**4 marks]**

The report should include a title page (including your name and ID number), 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. **[3 marks]**

Presentation

Note: If student would be unable to defend his/her coursework, s/he might be penalized with 50% of total coursework marks.

1. Marking Scheme

University Grading Scheme for Undergraduate Programmes: 2017/18		
Marking criteria	Letter grade	Mark recorded
C1 – Work Showing Evidence: *1. The report showed proper development process with* *a. Title page (Must use cover page provided by college with proper formatting)* *b. Table of contents (With page numbers)* *c. Table of figures (With page numbers)* *d. Introduction* *I. Introducing project content exceptionally well* *II. Must be short and descriptive* *e. Class diagram* *I. Must represent all classes* *II. Must be well formatted* *III. Position of variable, methods and data type must be correct* *IV. Well-structured* *f. Pseudocode* *I. Each method in all classes* *II. Writing one statement per line* *III. Capitalize initial keyword* *IV. Indent to show hierarchy* *V. End multiline structures (e.g. IF must be closed by ENDIF)* *g. Method descriptions* *I. Must be short and descriptive* *II. Exceptional writing and understanding level* *h. Evidence* *I. Screenshots as an evidence* *II. Each screenshots must display user desktop clearly as a proof for his/her own work.* *i. Testing* *I. Must include test cases clearly displaying expected result, test result and test completion status.* *II. Screenshots must be clear* *III. Screenshots must be well leveled* *j. Different error detection and correction* *I. Minimum four errors detection should be covered.* *II. Table for error detection cases clearly showing expected result, actual result and result status.* *III. Detected errors should be solved*	A+	95

IV. *Proof for above actions should be include in report as screenshots.* V. *Short description on errors* k. *Conclusion* I. *Evaluation of work* II. *What student learnt* III. *Difficulties encountered by student* IV. *How they overcame* l. *Appendix* I. *List of the code* II. *Well formatted* m. *Quality of writing and the presentation of the report* 2. *Software Development* a. *Classes* I. *Class name must be same as in question* II. *Camel case rule should be followed* III. *Implementation of encapsulation properly* b. *Attributes* I. *Attributes must be exactly same as specified in question* II. *Proper implementation of encapsulation* III. *Data type must be accurate* c. *Variables and Methods* I. *Must follow camel case naming convention* II. *Variable and method name should be suitable and meaningful.* III. *Implementation of encapsulation properly* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 4. *Conclusion* a. ***Exceptional overall work (no fault in whole process)***		

C2 – Work Showing Evidence:		
1. The report showed proper development process with *a. Title page (Must use cover page provided by college with proper formatting)* *b. Table of contents (With page numbers)* *c. Table of figures (With page numbers)* *d. Introduction* *I. Outstanding introduction* *II. Must be short and descriptive* *e. Class diagram* *I. Must represent all classes* *II. Must be well formatted* *III. Position of variable, methods and data type must be correct* *IV. Well-structured* *f. Pseudocode* *I. Each method in all classes* *II. Writing one statement per line* *III. Capitalize initial keyword* *IV. Indent to show hierarchy* *V. End multiline structures (e.g. IF must be closed by ENDIF)* *g. Method descriptions* *I. Must be short and descriptive* *II. Outstanding writing and understanding level* *h. Evidence* *I. Screenshots as an evidence* *II. Each screenshots must display user desktop clearly as a proof for his/her own work.* *i. Testing* *I. Must include test cases clearly displaying expected result, test result and test completion status.* *II. Screenshots must be clear* *III. Screenshots must be well leveled* *j. Different error detection and correction* *I. Minimum four errors detection should be covered.* *II. Table for error detection cases clearly showing expected result, actual result and result status.* *III. Detected errors should be solved* *IV. Proof for above actions should be include in report as screenshots.* *V. Short description on errors* *k. Conclusion* *I. Evaluation of work* *II. What student learnt* *III. Difficulties encountered by student* *IV. How they overcame*	A	85

l. *Appendix* I. *List of the code* II. *Well formatted* m. *Quality of writing and the presentation of the report* 2. *Software Development* a. *Classes* I. *Class name must be same as in question* II. *Camel case rule should be followed* III. *Implementation of encapsulation properly* b. *Attributes* I. *Attributes must be exactly same as specified in question* II. *Proper implementation of encapsulation* III. *Data type must be accurate* c. *Variables and Methods* I. *Must follow camel case naming convention* II. *Variable and method name should be suitable and meaningful.* III. *Implementation of encapsulation properly* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 4. *Conclusion* ***a. Outstanding overall work (no fault in whole process)***		
C3 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Introducing project content excellently* II. *Must be short and descriptive* e. *Class diagram* I. *Must represent all classes* II. *Must be well formatted* III. *Position of variable, methods and data type must be correct*	A-	75

IV. Well-structured f. Pseudocode I. Each method in all classes II. Writing one statement per line III. Capitalize initial keyword IV. Indent to show hierarchy V. End multiline structures (e.g. IF must be closed by ENDIF) g. Method descriptions I. Must be short and descriptive II. Excellent writing and understanding level h. Evidence I. Screenshots as an evidence II. Each screenshots must display user desktop clearly as a proof for his/her own work. i. Testing I. Must include test cases clearly displaying expected result, test result and test completion status. II. Screenshots must be clear III. Screenshots must be well leveled j. Different error detection and correction I. Minimum four errors detection should be covered. II. Table for error detection cases clearly showing expected result, actual result and result status. III. Detected errors should be solved IV. Proof for above actions should be include in report as screenshots. V. Short description on errors k. Conclusion I. Evaluation of work II. What student learnt III. Difficulties encountered by student IV. How they overcame l. Appendix I. List of the code II. Well formatted m. Quality of writing and the presentation of the report 2. Software Development a. Classes I. Class name must be same as in question II. Camel case rule should be followed III. Implementation of encapsulation properly b. Attributes I. Attributes must be exactly same as specified in question II. Proper implementation of		

encapsulation *III. Data type must be accurate* *c. Variables and Methods* *I. Must follow camel case naming convention* *II. Variable and method name should be suitable and meaningful.* *III. Implementation of encapsulation properly* *d. Proper indentation must be implemented* *e. Exceptional Coding style will be awarded* *f. Using appropriate keywords* *g. Displaying suitable and appropriate messages to the user when system ask.* *3. Testing* *a. Proper test cases with Before and After Screen shots (Total 4 testing)* *b. At least 4 different error detection and correction (Screenshots required)* *1. Conclusion* ***a. Excellent overall work (avoidable fault in whole process)***		
C4 – Work Showing Evidence: *1. The report showed proper development process with* *a. Title page (Must use cover page provided by college with proper formatting)* *b. Table of contents (With page numbers)* *c. Table of figures (With page numbers)* *d. Introduction* *I. Introducing project content nicely* *II. Must be short and descriptive* *e. Class diagram* *I. Must represent all classes* *II. Must be well formatted* *III. Position of variable, methods and data type must be correct* *f. Pseudocode* *I. Each method in all classes* *II. Writing one statement per line* *III. Capitalize initial keyword* *IV. End multiline structures (e.g. IF must be closed by ENDIF)* *g. Method descriptions* *I. Must be short and descriptive* *II. Grate writing and understanding level* *h. Evidence* *I. Screenshots as an evidence* *II. Each screenshots must display user desktop clearly as a proof for his/her own work.* *i. Testing*	B+	67

I. Must include test cases clearly displaying expected result, test result and test completion status. II. Screenshots must be clear III. Screenshots must be well leveled j. Different error detection and correction I. Minimum three errors detection should be covered. II. Table for error detection cases clearly showing expected result, actual result and result status. III. Detected errors should be solved IV. Proof for above actions should be include in report as screenshots. V. Short description on errors k. Conclusion I. Evaluation of work II. What student learnt III. Difficulties encountered by student IV. How they overcame l. Appendix I. List of the code II. Well formatted m. Quality of writing and the presentation of the report 2. Software Development a. Classes I. Class name must be same as in question II. Camel case rule should be followed III. Implementation of encapsulation properly b. Attributes I. Attributes must be exactly same as specified in question II. Proper implementation of encapsulation III. Data type must be accurate c. Variables and Methods I. Must follow camel case naming convention II. Variable and method name should be suitable and meaningful. III. Implementation of encapsulation properly d. Proper indentation must be implemented e. Exceptional Coding style will be awarded f. Using appropriate keywords g. Displaying suitable and appropriate messages to the user when system ask. 3. Testing a. Proper test cases with Before and After Screen shots (Total 4 testing)		

b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* ***a. Great overall work with minor mistakes***		
C5 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Introducing project content is fine* II. *Must be short and descriptive* e. *Class diagram* I. *Must represent all classes* II. *Position of variable, methods and data type must be correct* f. *Pseudocode* I. *Each method in all classes* II. *Capitalize initial keyword* III. *End multiline structures (e.g. IF must be closed by ENDIF)* g. *Method descriptions* I. *Must be short and descriptive* II. *Fine writing and understanding level* h. *Evidence* I. *Screenshots as an evidence* II. *Each screenshots must display user desktop clearly as a proof for his/her own work.* i. *Testing* I. *Must include test cases clearly displaying expected result, test result and test completion status.* II. *Screenshots must be clear* III. *Screenshots must be well leveled* j. *Different error detection and correction* I. *Minimum three errors detection should be covered.* II. *Table for error detection cases clearly showing expected result, actual result and result status.* III. *Detected errors should be solved* IV. *Proof for above actions should be include in report as screenshots.* V. *Short description on errors* k. *Conclusion* I. *Evaluation of work* II. *What student learnt* III. *Difficulties encountered by student* IV. *How they overcame*	B	63

l. *Appendix* I. *List of the code* II. *Well formatted* m. *Quality of writing and the presentation of the report* 2. *Software Development* a. *Classes* I. *Class name must be same as in question* II. *Camel case rule should be followed* III. *Implementation of encapsulation properly* b. *Attributes* I. *Attributes must be exactly same as specified in question* II. *Proper implementation of encapsulation* III. *Data type must be accurate* c. *Variables and Methods* I. *Must follow camel case naming convention* II. *Variable and method name should be suitable and meaningful.* III. *Implementation of encapsulation properly* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* ***a. Fine overall work with minor mistakes***		
C6 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Introducing project content is considerable* II. *Must be short and descriptive* e. *Class diagram* I. *Must represent all classes* II. *Position of variable, methods and data type must be correct*	C+	57

f. Pseudocode *I. Each method in all classes* *II. End multiline structures (e.g. IF must be closed by ENDIF)* *g. Method descriptions* *I. Must be short and descriptive* *II. Considerable writing and understanding level* *h. Evidence* *I. Screenshots as an evidence* *II. Each screenshots must display user desktop clearly as a proof for his/her own work.* *i. Testing* *I. Must include test cases clearly displaying expected result, test result and test completion status.* *II. Screenshots must be clear* *III. Screenshots must be well leveled* *j. Different error detection and correction* *I. Minimum two errors detection should be covered.* *II. Table for error detection cases clearly showing expected result, actual result and result status.* *III. Detected errors should be solved* *IV. Proof for above actions should be include in report as screenshots.* *k. Conclusion* *I. Evaluation of work* *II. What student learnt* *III. Difficulties encountered by student* *IV. How they overcame* *l. Appendix* *I. List of the code* *m. Quality of writing and the presentation of the report* *2. Software Development* *a. Classes* *I. Class name must be same as in question* *II. Implementation of encapsulation properly* *b. Attributes* *I. Attributes must be exactly same as specified in question* *II. Proper implementation of encapsulation* *III. Data type must be accurate* *c. Variables and Methods* *I. Variable and method name should be suitable and meaningful.* *II. Implementation of encapsulation*		

properly *d. Proper indentation must be implemented* *e. Exceptional Coding style will be awarded* *f. Using appropriate keywords* *g. Displaying suitable and appropriate messages to the user when system ask.* *3. Testing* *a. Proper test cases with Before and After Screen shots (Total 4 testing)* *b. At least 4 different error detection and correction (Screenshots required)* *1. Conclusion* ***a. Considerable overall work with some minor mistake***		
C7 – Work Showing Evidence: *1. The report showed proper development process with* *a. Title page (Must use cover page provided by college with proper formatting)* *b. Table of contents (With page numbers)* *c. Table of figures (With page numbers)* *d. Introduction* *I. Introducing project content is normal* *II. Must be short and descriptive* *e. Class diagram* *I. Must represent all classes* *II. Position of variable, methods and data type must be correct* *f. Pseudocode* *I. Each method in all classes* *II. End multiline structures (e.g. IF must be closed by ENDIF)* *g. Method descriptions* *I. Must be short and descriptive* *II. Normal writing and understanding level* *h. Evidence* *I. Screenshots as an evidence* *II. Each screenshots must display user desktop clearly as a proof for his/her own work.* *i. Testing* *I. Must include test cases clearly displaying expected result, test result and test completion status.* *II. Screenshots must be clear* *III. Screenshots must be well leveled* *j. Different error detection and correction* *I. Minimum two errors detection should be covered.* *II. Table for error detection cases clearly showing expected result, actual result and result status.*	C	53

III. *Detected errors should be solved* IV. *Proof for above actions should be include in report as screenshots.* k. *Conclusion* I. *Evaluation of work* II. *What student learnt* III. *Difficulties encountered by student* IV. *How they overcame* l. *Appendix* I. *List of the code* II. *Well formatted* m. *Quality of writing and the presentation of the report* 2. *Software Development* a. *Classes* I. *Class name must be same as in question* II. *Implementation of encapsulation properly* b. *Attributes* I. *Attributes must be exactly same as specified in question* II. *Proper implementation of encapsulation* III. *Data type must be accurate* c. *Variables and Methods* I. *Variable and method name should be suitable and meaningful.* II. *Implementation of encapsulation properly* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* a. ***Normal overall work with some major mistakes***		
C8 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Must be short and descriptive*	D+	47

e. Class diagram I. Must represent all classes f. Pseudocode I. Each method in all classes g. Method descriptions I. Must be short and descriptive h. Evidence I. Screenshots as an evidence II. Each screenshots must display user desktop clearly as a proof for his/her own work. i. Testing I. Must include test cases clearly displaying expected result, test result and test completion status. II. Screenshots must be clear j. Different error detection and correction I. Minimum one errors detection should be covered. II. Detected errors should be solved III. Proof for above actions should be include in report as screenshots. k. Conclusion I. Evaluation of work II. What student learnt III. Difficulties encountered by student IV. How they overcame l. Appendix I. List of the code II. Well formatted m. Quality of writing and the presentation of the report 2. Software Development a. Classes I. Class name must be same as in question II. Implementation of encapsulation properly b. Attributes I. Attributes must be exactly same as specified in question II. Data type must be accurate c. Variables and Methods I. Variable and method name should be suitable and meaningful. d. Proper indentation must be implemented e. Exceptional Coding style will be awarded f. Using appropriate keywords g. Displaying suitable and appropriate messages to the user when system ask. 3. Testing a. Proper test cases with Before and After Screen shots (Total 4 testing)		

b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* a. ***Normal overall work with many major mistakes***		
C9 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Must be short and descriptive* e. *Class diagram* I. *Must represent all classes* f. *Pseudocode* I. *Each method in all classes* g. *Method descriptions* I. *Must be short and descriptive* h. *Evidence* I. *Screenshots as an evidence* II. *Each screenshots must display user desktop clearly as a proof for his/her own work.* i. *Testing* I. *Screenshots must be clear* II. *Screenshots must be well leveled* j. *Different error detection and correction* I. *Minimum one errors detection should be covered.* II. *Detected errors should be solved* III. *Proof for above actions should be include in report as screenshots.* k. *Conclusion* I. *Evaluation of work* II. *What student learnt* III. *Difficulties encountered by student* IV. *How they overcame* l. *Appendix* I. *List of the code* II. *Well formatted* m. *Quality of writing and the presentation of the report* 2. *Software Development* a. *Classes* I. *Class name must be same as in question* b. *Attributes* I. *Attributes must be exactly same as specified in question* II. *Data type must be accurate*	D	43

c. *Variables and Methods* I. *Variable and method name should be suitable and meaningful.* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* ***a. Normal overall work with many major errors***		
C10 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Must be short and descriptive* e. *Class diagram* I. *Must represent all classes* f. *Pseudocode* I. *Each method in all classes* g. *Method descriptions* I. *Must be short and descriptive* h. *Evidence* I. *Screenshots as an evidence* i. *Testing* I. *Screenshots must be clear* j. *Different error detection and correction* I. *Minimum one errors detection should be covered.* II. *Proof for above actions should be include in report as screenshots.* k. *Conclusion* I. *Evaluation of work* II. *What student learnt* III. *Difficulties encountered by student* IV. *How they overcame* l. *Appendix* I. *List of the code* m. *Quality of writing and the presentation of the report* 2. *Software Development* a. *Classes* I. *Class name must be same as in question*	F1	37

b. *Attributes* I. *Attributes must be exactly same as specified in question* c. *Variables and Methods* I. *Variable and method name should be suitable and meaningful.* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* ***a. Poor overall work with major errors***		
C11 – Work Showing Evidence: 1. *The report showed proper development process with* a. *Title page (Must use cover page provided by college with proper formatting)* b. *Table of contents (With page numbers)* c. *Table of figures (With page numbers)* d. *Introduction* I. *Must be short and descriptive* e. *Class diagram* I. *Must represent all classes* f. *Pseudocode* I. *Each method in all classes* g. *Method descriptions* I. *Must be short and descriptive* h. *Evidence* I. *Screenshots as an evidence* i. *Testing* I. *Must include test cases clearly displaying expected result, test result and test completion status.* II. *Screenshots must be clear* j. *Different error detection and correction* I. *No errors detection* k. *Conclusion* I. *Evaluation of work* II. *What student learnt* III. *Difficulties encountered by student* IV. *How they overcame* l. *Appendix* I. *List of the code* m. *Quality of writing and the presentation of the report* 2. *Software Development*	F2	23

a. *Classes* I. *Class name must be same as in question* b. *Attributes* I. *Attributes must be exactly same as specified in question* c. *Variables and Methods* I. *Variable and method name are not suitable and meaningful.* d. *Proper indentation must be implemented* e. *Exceptional Coding style will be awarded* f. *Using appropriate keywords* g. *Displaying suitable and appropriate messages to the user when system ask.* 3. *Testing* a. *Proper test cases with Before and After Screen shots (Total 4 testing)* b. *At least 4 different error detection and correction (Screenshots required)* 1. *Conclusion* ***a. Very poor overall work with many major errors***		
Fail (non-submission or submission of work which cannot be given any credit (e.g., blank submission, incorrect assignment)	F3	0