**Project deliverables**

This project can be submitted by a team of 5 to 6 members from the same lab. Make sure all team member’s names and student numbers are recorded on the Team Cover Sheet. One person should be nominated as the group leader.

**Your submission must include the following**:  
 Cover Page (included);   
 Printout / Listing of your code (executable form);  
 Printout /Listing of the output file.  
 (Use the Print function in WORD, Notepad, or WordPad)  
 *Do Not use a copy and paste*;  
 Function Chart;  
 *Flowcharts are not required.*

**Program Requirements**

This project will use the same files as used in Project 2. However, the Student file (STUFILE.TXT) will first be converted into an Indexed Sequential file. The file for the Program Records will remain the same as a Line Sequential file.

**First program**. Convert the Student File into an Indexed Sequential file.  
**Second program**. Update the Indexed Sequential Student File with on-line (interactive) transactions using a Screen Section. The online transactions to be used will be provided. Note – you must use the Screen Section to handle the transactions needed to update data in records on the file. This program would prompt the user to enter a student number, then, search the file for that student number. If found, it would display the student information on screen for update. Then, accepts user update and updates the Indexed Sequential Student File.  
**Third Program**. Re-run the report program from Project 2 with the following amendments:   
 1) the Data Division component declaring the structure of the Program Table must be handled as a COPY member;  
 2) the code which calculates the average must be an externally executed program with a CALL function.

Both files (STUFILE.txt and PROGRAM.txt) will be provided.  
The transactions to be used in updating the Student File will be provided.  
Record structures for all files (input and output) are the same as for Project 2 but are repeated below for reference.   
  
The required Screen layout will be provided under separate cover.

**Input Record Structures**The structure for each student record is below. The file to be used to test your code will be provided with the file name STUFILE.txt for the student records, and PROGRAM.txt for the different Programs of Study. There are five courses for each student record. Each course record (1 through 5) contains the Course Code followed by the average for that course.   
 **Record structure -- Program File (PROGRAM.txt)  
 (** Note – there will be a maximum of 20 occurrences of records)PROGRAM CODE 6 bytes alphanumeric   
PROGRAM NAME 20 bytes alphanumeric  
(**Note** – This file is to be loaded into a table for use in the report program) **Record Structure -- Student Record (STUFILE.txt)** STUDENT NUMBER 6 bytes numeric   
 TUITION OWED 6 bytes numeric (including 2 assumed decimals)  
 STUDENT NAME 40 bytes alphanumeric..   
 PROGRAM OF STUDY 6 bytes alphanumeric   
 COURSE CODE 1 7 bytes alphanumeric  
 COURSE AVERAGE 1 3 bytes numeric  
 COURSE CODE 2 7 bytes alphanumeric  
 COURSE AVERAGE 2 3 bytes numeric  
 COURSE CODE 3 7 bytes alphanumeric  
 COURSE AVERAGE 3 3 bytes numeric  
 COURSE CODE 4 7 bytes alphanumeric  
 COURSE AVERAGE 4 3 bytes numeric  
 COURSE CODE 5 7 bytes alphanumeric  
 COURSE AVERAGE 5 3 bytes numeric

**Output Record Structure (**Student Report File)Each output record will be a structured on a single line as described below. You must use Each the ORGANIZATION clause as LINE SEQUENTIAL in the SELECT ASSIGN statement. STUDENT NAMEfiller (2 spaces)  
STUDENT AVERAGE (This field must be rounded to a whole integer)  
filler (4 spaces)  
PROGRAM NAME ( retrieved from the table)  
filler 4 spaces  
TUITION OWED (this field must be edited with inserted decimal point, commas as suppression of leading zeros)  
  
**Output Record Structure** (Column Header)  
 You must have a Column Header Record at the top of the report with the words NAME, AVERAGE, PROGRAM, TIUTION OWED in line with the beginning of each field.  
  
**Processing requirements Student Report file**The report program will handle the same process as that for Project 2. However, some changes will be required.  
1) That Cobol source code in the Data Division which declares the structure of the Program Table must be removed from the program code. Place (save) that code in an external file. In the Report Program, you must use the COPY command in your source code to access that code when your program is compiled.  
2) The code in the Procedure Division must be removed and set up as a separate program that will be accessed using the CALL command.  
3) Reference to the Indexed Sequential file for the Report Program will access the records in a sequential manner as with Project 2.

TEAM COVER PAGE

**NAME** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
STUDENT NUMBER \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
**NAME** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
STUDENT NUMBER \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
**NAME** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
STUDENT NUMBER \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
**NAME** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
STUDENT NUMBER \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
**NAME** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
STUDENT NUMBER \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
**NAME** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
STUDENT NUMBER \_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOTE If any student does not participate fully in the project, the group leader should identify that student and indicate what level of participation should be recognized and why. Use a range from zero (0) to ten (10)  
(0 -- no participation 5 --- partial participation 10 – full participation)

===============================================================

**Marking Scheme (as applicable)**

### Refer to Notes Regarding Grading below for specific points that will be checked and influence the marks allocated.

Output format and content \_\_\_ / 20  
Program listing \_\_\_ / 50  
Documentation \_\_\_ / 30

**TOTAL \_\_\_\_ / 100**

# Comments / feedback (to be made by instructor)

**Notes Regarding Grading**

The **program listing** will be examined primarily for:

1. relationship to function chart and flowchart;
2. use of prescribed commands as required by the problem specifications;
3. application of standards and structures;
4. use of proper functional constructs (cohesion and coupling);
5. use of internal comments (if required);
6. Successful compilation and execution.  
     
   The **output reports (hard copy or screen displays if required)** will be examined for accuracy of the output information and the prescribed format as noted in the program requirements.

The **documentation** will be examined to ensure:

1. proper use of symbols and logical diagrams/narratives (i.e. flowcharts);
2. proper structure and content of structure/function/hierarchy charts;
3. clear and accurate report or screen layouts (if required);
4. clear description or comments of the program logic; and,
5. relationship to the program code.   
     
   Any violation of the established standards (Standards document in the Course Information content area of Brightspace) will result in a loss of at least 5 points.  
   Any discrepancy regarding the grading notes above will result in a loss of at least 5 points.