COMP2611: Data Structures
Programming Assignment Project #2

DEADLINE DUE DATE: Sunday 14 April, 2024 - No Later Than Midnight.

Objective:

This project assignment is an extension of the first project assignment and will also be constructed with the usage of **wxWidgets** to develop a Graphical User Interface (GUI) in C++ with the Hierarchical Abstract Data Types (ADTs) as the main menu items and the member functions of the ADTs as sub-menu items. The partial C++ code for the Binary Search Tree (BSTree) class will be made available to you, but you will be required to complete the code and integrate it within your GUI. You will be required to develop the code for ALL the following ADTs:

1. Binary Search Tree (***Partial code provided***)
2. AVL Tree
3. Red-Black Tree
4. Splay Tree
5. Maximum Binary Heap
6. Minimum Binary Heap
7. Set

TASKS:

1. Develop the frame for your GUI with the following main Menu bar:

File	BS Tree	AVL Tree	RB Tree	Splay Tree	MaxHeap	MinHeap	Set	Help

2. The ADTs defined in the main menu bar should have sub-menus as follows:

File
Open File
Create ADTs
Add Data
Display File
Save File
Save As
Clear Window
Exit

BST
Inorder
Preorder
Postorder
Delete Data

AVL Tree
Inorder
Preorder
Postorder
Delete Data

RB Tree
Inorder
Preorder
Postorder
Delete Data

Set
Display SetA
Display SetB
Display Intersection
Display Union
Delete from SetA
Delete from SetB

Splay Tree
Inorder
Preorder
Postorder
Delete Data

MaxHeap
Add Data
Display All
Delete Data
Heap Sort

MinHeap
Add Data
Display All
Delete Data
Heap Sort

Help
About
Exit

3. The GUI should be captioned: "**COMP2611 – Students Registration Database.** The Status bar should be divided into three (3) sections with the string **"Ready…"** in **Section 1, "Assignment :** *<Name Surname>***"** in **Section 2** and your **ID Number** in **Section 3.**

e.g.

Ready…	Assignment: John Doe	123456789

METHOD:

Your program should be created using wxWidgets in Linux OpenSUSE!

Your project Application will read data from the ***binary file,*** "**StudentsData.dat**", which has been provided. However, **the file name should NOT be hard-coded into your program**. The file is constituted of records for each student, which represents students with majors in, Computer Science (COMP), Information Technology (IT), Electronics (ELET), Mathematics (MATH), and Physics (PHYS). The record structure within the file is constituted in the order as follows:

- An integer field (ID Number)
- A string of 15 characters (Name)
- A string of 20 characters (Surname)
- A string of 6 characters (Major)
- A floating point field (GPA).

Use the records in the data file to populate the ADTs of your project as follows:

1.	**BST**	**All records**
2.	**AVL Tree**	**Computer Science major**
3.	**RB-Tree**	**IT major**
4.	**Splay-Tree**	**Mathematics major**
5.	**MaxHeap**	**Electronics and Physics majors**
6.	**MinHeap**	**Mathematics major**
7.	**Sets**	**Set A – All students with GPA LESS than 3.0** **Set B – All students with GPA Greater than or equal to 3.0** **Intersection set – Computer Science majors**

In Addition:

1. The purist definition will be used for ALL the Binary Search Tree ADTs.

2. The **File** menu item should have the sub-menus of **Open File, Create ADTs, Add Data, Display File, Save File, Save As, Clear Window, and Exit**. When **Open File** is clicked, the **system fileOpen dialog** should be opened with the option to display files of type: **Data (*.dat) , Text (*.txt)** and **All (*.*).** The user must also be able to type in the file name in the filename textbox.

3. Once the contents of a file are displayed or the result of some processing is displayed, the menu selections of **Save File** and **Save As** should open the corresponding dialogs to perform the desired task. **Save As** should allow the user to specify a file name and file type into which the contents can be saved. These two functions should save **ONLY** the contents of the main textbox.

4. When the file is opened, its contents should be **immediately** displayed in that main textbox as well as when the menu option for **Display File** (in the menu: **File**) is clicked.

5. When the **Create ADTs** sub-menu item from **File** is clicked, all the ADTs should be populated with the contents of the **StudentsData.dat** file, as specified in the ADTs table above.

6. When **Add Data** is clicked, an input dialog should be opened to receive the data for the record which will be manually inserted. The new record should then be inserted into **ALL** the ADTs that it qualifies for, as specified in the ADTs table above.

7. When the **Display File** is clicked, ALL the records from the data file should be displayed one record per line, provided a file has been opened. The same applies when an ADT's sub-menu item requires that nodes be displayed.

8. The operation indicated by a sub-menu item should then be carried out on that particular ADT **ONLY**. An ADT **should NOT** be affected by the operations carried out on another ADT.

9. The nodes in the AVL tree must contain an attribute to describe the weight of the node (i.e. negative for **left-heavy**, positive for **right-heavy,** and zero for **balanced**), which must be displayed in square brackets at the end of the line when the AVL tree is traversed.

10. The nodes of the Red-Black tree must contain an attribute to describe the node's color, which must also be displayed in square brackets at the end of the line when the Red-black tree is traversed.

11. For all the **tree ADTs**, when their nodes are displayed, the ID numbers of the left and right children should also be indicated in square brackets as [***left, right***]. If there is no left and/or right child, then ***NULL*** should be used in the appropriate space. You should use the display output provided in the BST code, as your guide.

12. The **Help** menu item should contain sub-menu items for **About** and **Exit**. The About sub-menu event should display a dialog box with an appropriate caption and the student's name, ID number, and program version number in the body of the dialog box, as well as an OK button. The **Exit** event should be the same as in the **File** menu item.

13. The **Exit** menu item is to close the program.

14. When the cursor is placed on a sub-menu item, a description of the menu option should be displayed in the first partition of the status bar. At all other times, the string, "**Ready…**" should be displayed.

15. A record should be deleted from a specific ADT by providing only the **ID Number** to an input Dialog box.

SUBMIT:

Your project should contain a separate header file for each of the ADTs (i.e. BSTree.h, AVLTree.h, RBTree.h, SplayTree.h, MinHeap.h, MaxHeap.h, and Set.h) as well as a source code file (**.cpp**) which contains your main program code. All the files should be submitted as a

ZIP file with **YOUR ID number as the name** (e.g. **123456789.zip**) through the portal in this course's page on eLearning. You may also use the portal to continually save your zipped project as a ***DRAFT*** as you are developing your code. Previous versions of your submissions will be overwritten and only the latest version will be kept.

You may use any version of Linux to create your code but **double-check** that your program code can be compiled in **OpenSUSE on the machines in CSL2**.

DEADLINE DUE DATE :

Sunday 14 April, 2024 - No Later Than Mid-Night.

No late assignments will be accepted – No extension will be granted!