**Chapter 15: Object Oriented Programming - continued**

Continuing from last week and taking what you learnt, you have one last program to create. Read the instructions carefully!

**Program 41: Bank Account**

Write a program, which presents the user with the following menu:

1. Open a new account

2. View an account

3. Close an account

4. Exit program

Upon entering ‘**1**’ this second account menu is presented:

1. General account

2. Junior account

3. Savings account

4. Return to main menu

See below for the actions to be carried out when selecting from this menu.

Upon entering a ‘**2**’ the user is asked to enter an account number and if the account exists the following details are presented: Account number, Surname, Forename, Address, Balance, Interest rate and Account type. The user is then returned to the main menu.

Upon entering a ‘**3**’ the user is asked to enter an account number and if the account exists the account is closed. Closing an account removes it from the save file.

Upon entering a ‘**4**’ the program closes.

Within the account menu if the user enters a ‘4’ they should be returned to the main menu. On any other selection they are taken through a process of entering their details. These are to include: Surname, Forename & Address. They should be asked how much money they wish to place into the account to open it – this must be greater than 0 otherwise the process is cancelled. A unique account number is **provided** for the user. When the account has been opened the user is returned to the main menu.

All details must be stored on file so that they can be accessed between executions of the program. Ensure that adding a new account does not corrupt the already saved data. Likewise, when closing an account.

To complete this task, you should use the object-oriented approach described in this chapter. There should be a base account class with a junior account and savings account which both inherit from it. The differences in accounts will be the interest rates and account type.

Account type should be an **enum** consisting of GENERAL, JUNIOR and SAVINGS.

Provide screenshots of all menus and actions, along with the save file. The data shown in the file must correspond to the screenshots demonstrating it being used.

**Program 41 Source code (by file):**

**BaseAccount.h:**

****

**BaseAccount.cpp:**

****

**JuniorAccount.h:**

****

**JuniorAccount.cpp:**

****

**SavingsAccount.h:**

****

**SavingsAccount.cpp:**

****

**Main.cpp:**

****

****

****

****

**Program 41 Screenshots of output:**

**Program Account Save file:**