Project for Object-Oriented Programming

Project Deliverables

The source code for this project are the following files:

> `ProjPartA.cpp` – your source code program for Part A
> `ProjPartB.cpp` – your source code program for Part B

The **grade** of your project is determined by the followings:

– Design with class diagrams (20%)

– Programming Style (20%): Style and efficiency of your program (comments are expected in the source code).

– Correctness (30%): We will verify the correctness of your program by testing it through

– Final Report (30%): The report includes the features of your system, a brief user manual of your system as well as the source codes of your system.

Specifications

1. The World Health Organization (WHO) compiles data about immunization levels around the world. The file named "`measles.txt`" contains data about the level of measles vaccinations in various countries over time.

Each line of the file contains the following fields, where there is one space between fields:

> Country (50 characters)
> Income Level (6 characters)
> Percent Vaccinated (3 characters)
> Region (25 characters)
> Year (4 characters)

The "Country" field contains the name of the country.

The "Income Level" field identifies the category assigned to that country by the World Bank:

> `WB_LI`   low income
> `WB_LMI` lower middle income
> `WB_UMI` upper middle income
> `WB_HI`   high income

The "Percent Vaccinated" field contains an integer number representing the percentage of children in that country who have received measles vaccine by the age of one.

The "Region" field identifies the region assigned to that country by WHO.

The "Year" field contains the year for which the data was compiled.

2. You will develop the two programs described below. Each program will be in a separate source code file (the names are given under "Project Deliverables").

Part A

1. The program in "ProjPartA.cpp" will copy selected lines from "measles.txt" into a file selected by the user.

a) The program will always read from "measles.txt" (it will not prompt the user for the name of the input file). If it is unable to open that file, the program will halt.

b) The program will prompt the user for the name of the output file. If that file does not exist, the program will create it and continue. If that file does exist, the program will discard the current contents of the file and continue.

c) The program will prompt the user to enter a year, and will copy all lines of "measles.txt" selected by the user's response. A line is selected if the user's response matches the Year field or any of its prefixes. All lines are selected if the user's response is any of the values in the set {"", "all", "ALL"}. Note that "" is the empty string.

For example, a line whose Year field contains "1987" would be selected by any of the following user responses: {"1", "19", "198", "1987", "", "all", ""ALL"}.

2. The output file created by the program will have the same format as the input file (same field widths and spacing). Note that when the user selects all lines, the output file will be identical to the input file.

3. The program will display appropriate messages to inform the user about any unusual circumstances.

Part B

1. The program in "`ProjPartB.cpp`" will display one summary report to the user.

a) The program will prompt the user to enter the name of the input file. If it is unable to open that file, the program will prompt the user again until the user enters a valid file name.

b) The program will prompt the user to enter a year, and will then prompt the user to enter an income level. The income level must be one of the characters in the set {1, 2, 3, 4}, where 1 corresponds to "low income", 2 corresponds to "lower middle income", 3 corresponds to "upper middle income" and 4 corresponds to "high income".

c) The program will identify all records (lines) in the input file which match the user's criteria for year and income level, and the program will display a report with the following information:

    The count of records in the input file which match the user's criteria
    The average percentage for those records (displayed with one fractional digit)
    The country with the lowest percentage for those records
    The country with the highest percentage for those records

The name of the country and the percent of children vaccinated will be displayed for the last two items (lowest percentage and highest percentage).

2. The program will display appropriate messages to inform the user about any unusual circumstances.

3. The program will contain the following functions (you may develop additional functions):

```
open_file()  → file object
process_file( file object )  → None
```

The notation above gives the name of each function, the number and type of its argument(s), and the type of its return value.

a) The function names will be spelled exactly as shown (for example, `open_file`).

b) Function `open_file` has no parameters. It returns a file object after prompting the user to enter the name of the input file (see above).

c) Function `process_file` has one parameter (a file object). It performs the processing to read the input file and display the report.