Assignment 1 - part 2

Deadline	**Anytime before Sunday, 1st of April 2018, 15:00**
Evaluation	10 marks –which is 5% of your final grade
Late Submission	5% per hour (or fraction of hour) it is late
Team	The assignment can be done individually or in pairs (of at most 2 students)
Purpose	Practice with C++ input and output, basic constructs and structures.

Problem to solve

INMS admin was very happy with the program you made in the previous assignment. Your program helped them identify invalid requests very quickly. However, with so many requests made for one day, even that is not enough. They politely ask you to make the program fully automated. All booking requests that do not specify a room, have the wrong purpose, or exceed the time allowed, will be automatically reassigned a room with the appropriate purpose and time. **Be aware that no room can be booked for a total of more than 300 minutes each day**.

As before, room descriptions are stored in a file name `rooms.txt`, and requests for bookings are in `bookings.txt`. Your program will read these two files, detect invalid booking requests, reassign rooms to these requests, and print out the new arrangement.

In this assignment, you will be using classes instead of structs. You will be given a header file that includes class prototypes and all necessary methods declared. Some of them are already implemented. You will implement the remaining methods according to the method declaration.

Name your program `a1p2.cpp`
Hand-in: Submit `a1p2.cpp` electronically using the submission form on STREAM.

If you have any questions about this assignment, please ask the lecturer before the assignment is due.

Problem statement

Write a program that does the following:

1. Displays on the standard output information about the authors of the program (IDs , family & given names, assignment number)
2. Reads data contained in the files `rooms.txt` and `bookings.txt`, as necessary.
3. If there is any problem reading the files, stops immediately and outputs an error message to the standard error stream.
4. If both files are read successfully, search for invalid requests and assign then the appropriate rooms and time.

As a reminder, invalid booking requests are those which:

1. Do not have the same purpose as the room they book, or
2. Book longer than the time allowed by the room, or
3. Do not specify a room.

Input and output

Input

Input files are in the same format as in assignment 1 part 1. You can assume that all the requests can be accommodated (i.e. there will not be too many requests that all rooms run out of the 300-minute limit.)

Sample output

Given the same sample input as A1P1, your program should produce the following output- make sure the **formatting** of your output is **well structured** and follows **EXACTLY** the one here:

```
************************************************
* 159.234 Assignment 1 Part 2
* Name: Jane Doe, John Smith
* ID: 17012345, 16234567
************************************************
Initially, staff made the following booking requests:
Booking: by cscorgings for teaching purpose at room INMS-Lab1 for 60 minutes
Booking: by ecalude for teaching purpose at room INMS-Lab1 for 60 minutes
Booking: by dpplayne for teaching purpose at room INMS-Lab2 for 120 minutes
Booking: by agilman for meeting purpose at room INMS-Lab2 for 60 minutes
Booking: by ecalude for meeting purpose at room INMS-3.08 for 180 minutes
Booking: by jren for teaching purpose at room INMS-3.08 for 120 minutes
Booking: by tliu for teaching purpose at room --- for 60 minutes
Booking: by mpawley for teaching purpose at room --- for 60 minutes

Now fixing invalid requests...

Final booking arrangement is:
Booking: by cscorgings for teaching purpose at room INMS-Lab1 for 60 minutes
Booking: by ecalude for teaching purpose at room INMS-Lab1 for 60 minutes
Booking: by dpplayne for teaching purpose at room INMS-Lab2 for 120 minutes
Booking: by agilman for meeting purpose at room INMS-2.14 for 60 minutes
Booking: by ecalude for meeting purpose at room INMS-3.08 for 120 minutes
Booking: by jren for teaching purpose at room INMS-Lab2 for 120 minutes
Booking: by tliu for teaching purpose at room INMS-Lab1 for 60 minutes
Booking: by mpawley for teaching purpose at room INMS-Lab1 for 60 minutes

Room utilisation is:
Room: INMS-Lab1 suitable for teaching purpose allows maximum 60 minutes. Total booked time: 240 minutes.
Room: INMS-Lab2 suitable for teaching purpose allows maximum 120 minutes. Total booked time: 240 minutes.
Room: INMS-2.14 suitable for meeting purpose allows maximum 60 minutes. Total booked time: 60 minutes.
Room: INMS-3.08 suitable for meeting purpose allows maximum 120 minutes. Total booked time: 120 minutes.
```

If for example, file `rooms.txt` is missing, your program must print out EXACTLY[1] the following:

```
************************************************
* 159.234 Assignment 1 Part 2
* Name: Jane Doe, John Smith
* ID: 17012345, 16234567
************************************************
Error: file rooms.txt is missing. Program terminates.
```

Your program will be compiled, run and evaluated by a machine. So besides your name(s) and ID(s), the output must be EXACTLY the same, character to character.

Requirements (read carefully because they're different from A1P1. The most important points are **in bold)**

What you must and must not do

1. Your program MUST be named `a1p2.cpp`. If you use a different name, you will get zero.

2. You MUST NOT submit anything other than `a1p2.cpp`.

3. **You MUST use only the classes provided in `a1p2_classes.h`. No extra class can be declared.**

4. **You MUST include the provided header using `#include "a1p2_classes.h"`. DO NOT copy its content into `a1p2.cpp`.**

5. **You MUST NOT make any changes to the header file.**

6. **You MUST NOT include any non-C++ standard library header files. E.g. `window.h` is a Windows-specific header and you MUST NOT include it.**

7. **You MUST NOT use the `exit` function. The only way you may terminate your program is to `return` from the `main` function.**

8. **You MUST follow the same style and format that are used in the header.**

9. You MUST read the information from the input files into an array of Bookings and an array of Rooms. These arrays must not be global.

10. Each file MUST only be read ONCE.

[1] Of course your names and IDs should be used instead of Jane Doe…..

This means you are not allowed to repeatedly read from the input files over and over in order to perform the consistency checks.

11. Your program MUST NOT require any user interaction.

This means you cannot ask the user to press any key, or type in anything. Your program should read the input files and perform the consistency check automatically. When your program finishes printing out the inconsistency, TERMINATE IMMEDIATELY.

12. Your program SHOULD be organised as followed:
    - Header (Your name(s), ID (s), short description for the program, etc)
    - All included files/libraries
    - Main function
    - Class methods' definition

13. **All class methods (as provided in the header) have already been properly documented. But any extra functions you make MUST be properly documented following Doxygen format.**

14. **You MUST NOT use advanced features not covered in the course by the due date.**

Miscellaneous

1. When working in pair, send one solution file per team.

2. The assignment will be discussed on Wednesday lecture before the assignment is due and solutions will be discussed on Monday lecture after the due time.

3. Marks will be allocated for: correctness, completeness, use of C++ constructs presented in class/tuts, simple and clear solution, good documentation, and structured output display (on screen).

4. Using goto, non-constant global variables or C-like I/O constructs (i.e printf fprintf, scanf, FILE*,etc) is not allowed and it will be penalised. Only const global variables are allowed.

5. Programs that do not run or do not compile in the (Albany) labs, using gcc(SciTe), get 0 marks.

6. The program must be your own work. Please be aware that you might be asked to explain to your lecturer how your program works. If you cannot explain it, then it is not yours and you will get 0 marks for that assignment. Attributing someone else's work as your own is plagiarism, and it is a violation of Massey University policy. We might file an official complaint against any student who we believe has committed plagiarism.

7. Suspicious similar solutions will all get 0 marks-see also point 7 above