MOD003197 OBJECT ORIENTED C++ 2019-20 TRI2 MULTIPHASED COURSEWORK

IN-CLASS EXERCISES (15%)

During the trimester, weekly practical classes will include a number of short programming exercises (typically 8) that should be demo'd to the tutor in-class (the choice and number of exercises to be assessed is at the discretion of the tutor at the local point of delivery). Each successful demo is worth one mark. If 8 exercises are specified for assessment but only 6 are successfully demo'd, you will be awarded 6/8 th's of 15% towards the final mark. NB no work should be submitted for this component, only demo'd to the tutor during scheduled practical class time.

MAIN COURSEWORK ASSIGNMENT (85%)

A MARINA BERTH BOOKING SYSTEM

A new coastal marina has been built. The marina is a man-made single narrow canal in which boats can be moored one-in-front of another, but they cannot pass each other (this means that a boat or boats berthed nearer the entrance must be temporarily moved into a separate holding bay to allow a boat that wishes to leave access out of the marina – think of the holding bay as being a kind of separate lake, with no facilities, whose entrance is shared with the entrance/exit to the marina). The marina manager has contracted you to write a console-based program that will record berth bookings for the marina as each new boat arrives. On execution the program's main menu should first give the user four options; 1) record a new booking 2) delete a record 3) display all records (and available marina space) and 4) exit the program.

For new bookings you will have to check that the boat length is short enough (maximum length 15m) and that it is shallow enough (maximum draft 5m) and that there is sufficient room in the marina (the length of the marina is 150m). If the boat is too big or there is insufficient room you should terminate this transaction with an appropriate message and return to the main option menu. If the booking can proceed then the duration of stay in months should be requested and this is used to display the full price of the berth which is based on the length of the boat and on the duration of stay (10 pounds per meter per month). The user then has the choice to reject and quit the program or accept the offer. If the user is happy with the price the following information should be requested; a) Name of owner b) Name of boat c) Type of boat (narrow, sailing, or motor) and d) Boat length. This information should then be added to the system records, and the program returned to the main option menu.

Guidance hints

The complexity of a solution largely depends on functionality and what object-oriented principles you choose to implement. It is important to appreciate that this assignment can be treated purely as a booking system but can also be extended to be a simulation of the marina, at least as far as how boats are moved in and out of the marina and holding bay.
For example;

- You may choose to focus on the main operations of the marina, rather than data validation. Try to ensure there is no ambiguity on what to input.

- The berth records would ideally be stored as objects in a data structure such as an array or linked list (better). A non-OO approach is certainly possible but you will not get many marks for this approach.
- Each object in the data structure could be written to file on terminating the program and read from file on first starting the program (rather than having records that exist only for the duration of program execution).
- You could use inheritance to implement the different types of boat.
- You will have to think carefully about how boats are removed from the marina – the simplest solution is to choose *not* to implement deletions from the list, or merely delete the record from the list, but a much more sophisticated (and therefore technical) solution is to model the transient process of moving boats into the holding bay and back again.

Note that your program ideally should follow good object-oriented practice (e.g the main method should be minimal and only coordinate program execution, and you should use appropriate classes with suitably scoped attributes and methods). Ideally you should develop your code from principles as taught in class, but should you find and use any code which is from the public domain you **must** ensure it is properly attributed (referenced in the code itself by suitable comments and relevant reference sources such as URL's or other citations included in the report).

SUBMISSION AND ASSESSMENT

This work should be submitted by the Friday of teaching Week12 (the last week of teaching); for Cambridge this will be to the iCenter by 2pm on **Friday 24th April, 2020** (or by local arrangement for associate/partner colleges). The report should comprise a hard-copy print of all components i-iv below with an attached removable storage (DVD/CD/USB). The report itself should include the following>

i) A 500 word (max) evaluation of your approach to the problem (the design and functionality of your program especially the data structure and what could be done better). Whilst functionality should be briefly evaluated note you do *not* need to provide a separate section on formal testing of your program (no test plan and associated results needed).
ii) A class diagram of your program.
iii) Instructions on how to run your program.
iv) A listing of the C# source code.
v) An electronic copy of the report including the C# source code and also a *compiled executable copy* of the program on disk or USB stick.

For details of marking see the feedback sheet.

REASSESSMENT DETAILS

If you are unfortunate enough to require a re-assessment for this module coursework (normally as second attempt but possibly as a first attempt), you should re-submit the main coursework assignment (the marina booking system). Your mark out of 100% will then be calculated on this new submission only (usually the system will cap this at 40%); note there will be NO in-class exercise component taken into account irrespective of how well that component was originally undertaken. You are encouraged to explore a different approach to achieving a solution to the main assignment compared to the first attempt, if applicable.

MOD003197 feedback mark sheet 2019-20 2019-20 Tri2										
	In-class exercises (not used in resubmission)	**Main Assignment**							**Overall 010 mark**	**Overall module result**
Student	**/15**	**Functionality**	**OO design and code development**	**Documentation**	**Comment**	**%**	**Grade**	**/85**	**%**	**Grade**
	Weighted 0.15 Eg if 5 exercises are successfully demo'd out of 8 then the mark out of 15 is> 5/8 of 15 = 9.4	Reflects how well the program works e.g. appropriate user interface, holding bay usage, information display during processing, reading/writing from file.	Choice of user-written classes and their methods. Use of private data and public methods. Syntaxically accurate and appropriately detailed class diagram.	Presentation. Code clarity and layout. Correctness of class diagram. Evaluation of approach including success and limitations.	Personalized comments specific to student submission.	Each component is assigned a grade (F, D, C, B, A) and each maps to a mark (0-100) from which the average is taken.	Main assignment grade	Weighted 0.85	Overall weighted mark	F (<40); D (40-49); C (50-59); B (60 - 69); A (70+)
Sid Number	**Mark**	**Grade**	**Grade**	**Grade**	**Comment**	**Mark**	**Grade**	**Mark**	**Final Mark**	**Final Grade**