COS 120 – C++ Programming – Fall 2022

Homework 4

**If your Student ID is an even number, solve Task 1, if your Student ID is an odd number, solve Task 2.**

**Deliverables (via)**

1. Zipped **project folder** of your program.

2. Demonstration to me, on demand.

**Notes**

* Think carefully about the code in your programs – variable names, comments, etc.
* Programs must be fully commented – add a comment in the beginning describing what your program does and how! Also add any other comments where necessary.

**Deadline & Notes**

The deadline for the homework can be checked in canvas.

**Late work will be accepted but penalized**.

**Task 1 – Contacts**

Create an abstract class **Contact,** containing the **private** data items **name, number** (this should be a long int),and **email.**  Add an abstract method **call().** Also provide getter/setters for all the data items. Make **validations** so that **no invalid** values are entered (**number** should be a positive number).

Create a class **BlockedContact**, to inherit the **Contact** class. This class should contain one additional **private** data item – **blocked** – a boolean value – showing whether the contact is currently blocked or not. Include getter and setter for the additional data item, constructors and implementation of **call()**method to print “*Currently blocked, cannot make a call.*” if the **blocked** data items is true and “Calling xxx”, where xxx is the **name** of the contact if it is not blocked. Include a **friend** function to overload **operator>>** to read the name and number from the user and to update the data items.

Create a class **FavouriteContact** to inherit **Contact** class. This class should contain one additional **private** data item – **order –** that is showing the order of the contact in the favourites list. Include getter and setter for the **order**, **order** should be a number between 1 and 5. Implement **call**() to print “*Calling my xxx favourite contact - yyy*”, where xxx is the **order** and yyy is the **name** of the contact. Overload inequality operator (**operator!=**) to return true if **name** of the left operand is not the same as the **name** of the right operand.

Finally create a main function that demonstrates the creation of objects, the polymorphic function and operators that you have overloaded.

**Task 2 – Books**

Create an abstract class **Book,** containing the **private** data items **name**, **author** and **pages.**  Add an abstract method **type().** Also provide getter/setter for all the data items. Make **validations** so that **no invalid** values are entered – pages should be a positive number. Also overload the equality operator (**operator==**) to return true if the name and author of the left operand are the same as the name and author of the right operand.

Create a class **ChildrenBook**, to inherit the **Book** class. The **ChildrenBook** class should contain one additional **private** data item – **ageAppropriate** – the minimum age of its readers, include getter/setter for that data item, provide validations. Implement **type()** method to print *“Children book for xxx and older”*, where xxx is the value of the data item **ageAppropriate**. Include a **friend** function to overload **operator>>** to read the **name**, **author,** **pages** and **ageAppropriate** of the book and update the data items of the **ChildrenBook**.

Create a class **SeriousBook**, to inherit the **Book** class. The **SeriousBook** class should contain one additional **private** data item **genre**, include getter and setter for it. Implement **type()** method to print *“Serious book, genre: xxx”*, where xxx is the value of the data item **genre**.

Finally create a main function that demonstrates the creation of objects, the polymorphic function and operators that you have overloaded.