# Comp 4603

# Advanced C++

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment | 8 | Part | 2 |

Student Name: Alisher Shamayev

BCIT ID: A01182685

To implement **three** Behavioral design patterns using C++

Hint: It’s your choice on which three patterns

Submission:

1. Complete this document with pattern names with UML
2. C++ code/files
3. To submit one zip file including all your files

Late submissions will NOT be accepted.

1st design pattern name:

|  |
| --- |
| Iterator |

Description of your example:

|  |
| --- |
| In this case we made as class of school example so we have implemented Iterator design pattern which illustrates as StudentIterator interface, and VectorStudentIterator implementation of interface and last but not least Student concrete class. So in future we can make StudentList class and implement VectorStudentIterator method iterating through a list. Without changing in future of any concrete class or getting anything from concrete class. |

UML diagram

|  |
| --- |
| A piece of paper with writing on it  Description automatically generated |

2nd design pattern name:

|  |
| --- |
| Visitor |

Description of your example:

|  |
| --- |
| Here we have abstract class Visitor and two sub classes Client and Driver that will implement Visitor class method visit. Also we have Taxi concrete class that will call visit method in its own method. |

UML diagram

|  |
| --- |
| A piece of paper with writing on it  Description automatically generated |

3rd design pattern name:

|  |
| --- |
| Observer |

Description of your example:

|  |
| --- |
| Observer is abstract class and has own sub class Query. Other concrete base class is Database it has own several methods however it will call update method in its class through Query class and so we will see when and where the updates occurs through this design pattern. |

UML diagram

|  |
| --- |
| A piece of paper with writing on it  Description automatically generated |