**Object Oriented Programming**

**Lab report: 14**



|  |  |
| --- | --- |
| Name | Tayyaba |
| Reg no | FA21-BCE-093 |
| Class | BCE- 4 |
| Instructor’s Name | Prof. Sadaf Iqbal |

**Lab 14 – *Exception Handling***

**Lab Tasks**

**5.2.** Create a class which only works for absolute numbers, if it encounters any negative occurrence, then it throws an exception to its handler and display errors.

**Program:**

**Classes:**

A screenshot of a computer code

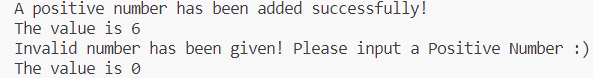
Description automatically generated with medium confidence

**Main:**

A screen shot of a computer code

Description automatically generated with low confidence

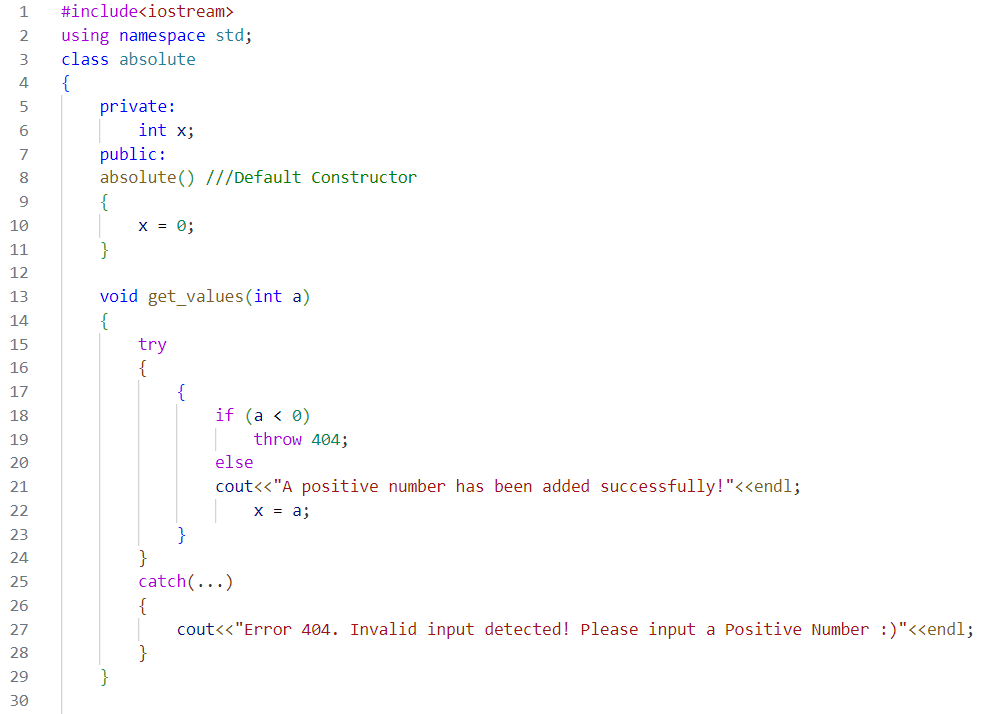
**Output:**



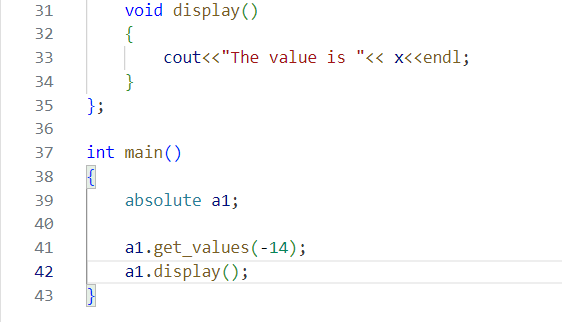
**5.3.** Modify the above task, by creating an exception class with an error code and corresponding error message. Code and message should be thrown and displayed in catch block.

**Program:**

**Classes:**



**Main:**



**Output:**



**Home Task:**

**6.1.** The *queue* is another data structure. A physical analogy for a queue is a line at a bank. When you go to the bank, customers go to the *rear* (end) of the line and customers who are serviced come out of the line from the *front* of the line. The main property of a queue is that objects go on the *rear* and come off of the *front* of the queue.

 *Make-Queue* Create a new, empty queue object.

 *Empty* Reports whether queue is empty or not.

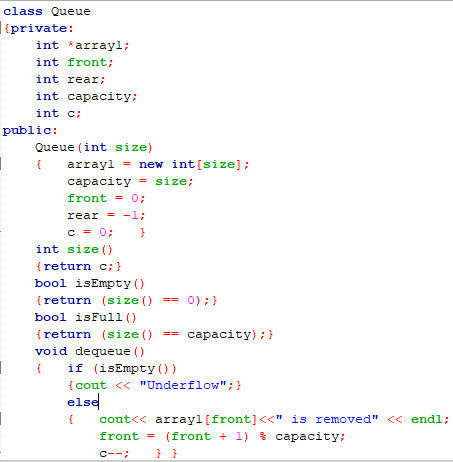
 *Enter(or Insert)* Places an object at the rear of the queue

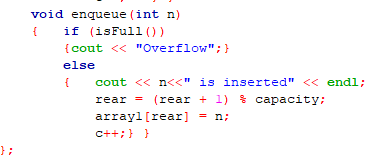
 *Delete (or Remove)* Removes an object from the *front* of the queue and produces that object.

Write a program to create a queue class and do queue operations with exception handling.

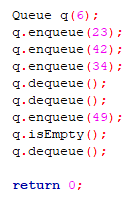
**Program:**

**Classes:**

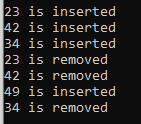




**Main:**



**Output:**



**Critical analysis:**

In this lab, we came to know that when executing C++ code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things. When an error occurs, C++ will normally stop and generate an error message. The technical term for this is: C++ will throw an **exception** (throw an error). Exceptions provide a way to transfer control from one part of a program to another. C++ exception handling is built upon three keywords: try, catch, and throw. It makes your programs more reliable and robust by dealing with exceptional conditions.