Object-Oriented Programming (3190)

Homework 8

Spring 2023

1. [10pt] Define a ′***reverse***′ function to reverse the order of elements in an array of any type. Implement a ′***swap***′ helper function to swap any two elements. Additionally, create a ′***print***′ helper function to display the contents of the arrays before and after swapping process.

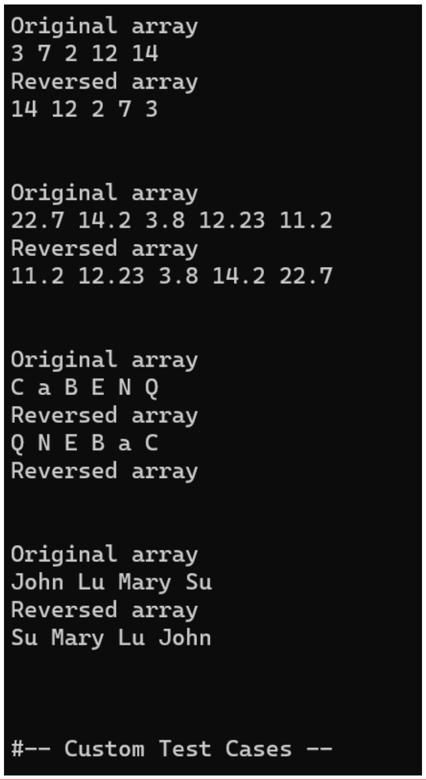
Your solution should fulfill the following requirements:

(a) Implement the ***reverse***, ***swap***, and ***print*** functions as template functions. The ***reverse*** function must use the ***swap*** function.

(b) Ensure that the output matches the provided sample output with arrays of integers, doubles, characters, and strings.

(c) Test your program with at least two additional test cases, utilizing different data types apart from int, double, char, and string.

Possible Output:



```
Original array
3 7 2 12 14
Reversed array
14 12 2 7 3


Original array
22.7 14.2 3.8 12.23 11.2
Reversed array
11.2 12.23 3.8 14.2 22.7


Original array
C a B E N Q
Reversed array
Q N E B a C
Reversed array


Original array
John Lu Mary Su
Reversed array
Su Mary Lu John



#-- Custom Test Cases --
```

Object-Oriented Programming (3190)
Homework8

2. [10pt] Create a templated class called "***Array***" that can handle an array of objects of any type and any size in the heap (using '***new***'). Define an "***add***" member function to add elements to the end of the array. Define a "***print***" function to display all elements in the array.

Your solution should fulfill the following requirements:

(a) Define a parameter constructor to set the capacity of the array (The maximum size of the array). Define a destructor to release the array.

(b) If the array is already full when adding an element, handle the exception using the '***out_of_range***' class.

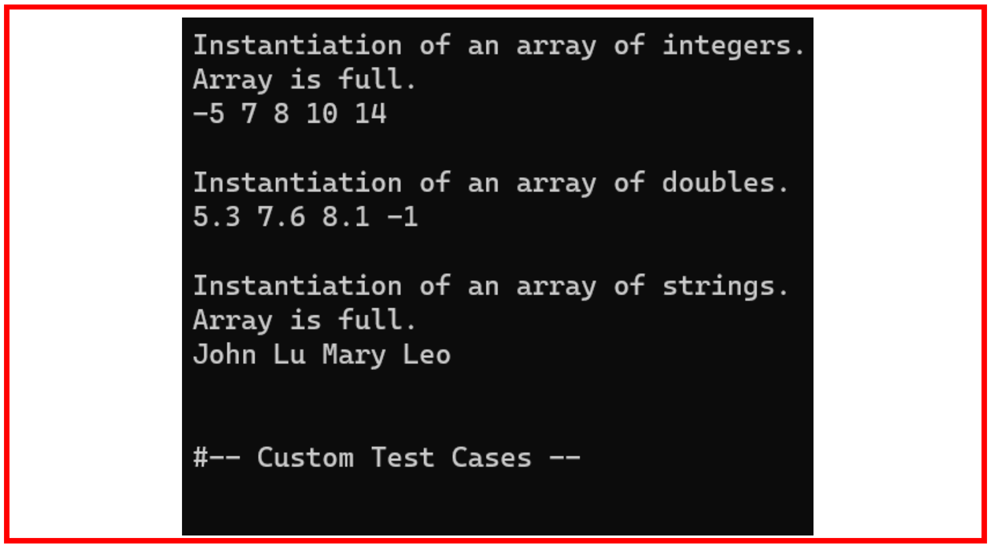
(c) Ensure that your program's output matches the provided sample output, considering the application file.

(d) Test your program with at least two more test cases, utilizing data types other than int, double, and string.

Application File:

```
int main()
{
        cout << "Instantiation of an array of
integers." << endl;
        Array <int> array1(5); // set the size
of the array using 'new'
        try
        {
                array1.add(-5);
                array1.add(7);
                array1.add(8);
                array1.add(10);
                array1.add(14);
                array1.add(20);
                array1.add(-14);
        }
        catch (out_of_range& ex)
        {
                cout << ex.what();
        }
        array1.print();

        cout << "Instantiation of an array of
doubles." << endl;
        Array <double> array2(5); // set the
size of the array
        try
        {
                array2.add(5.3);
                array2.add(7.6);
                array2.add(8.1);
                array2.add(-1);

        }
        catch (out_of_range& ex)
        {
                cout << ex.what();
        }
        array2.print();
        cout << "Instantiation of an array of
strings." << endl;
        Array <string> array3(4); // set the
size of the array
        try
        {
                array3.add("John");
                array3.add("Lu");
                array3.add("Mary");
                array3.add("Leo");
                array3.add("Robert");
        }
        catch (out_of_range& ex)
        {
                cout << ex.what();
        }
        array3.print();


        cout << "\n#-- Custom Test Cases --
\n\n";

        return 0;
}
```

Possible Output:



```
Instantiation of an array of integers.
Array is full.
-5 7 8 10 14

Instantiation of an array of doubles.
5.3 7.6 8.1 -1

Instantiation of an array of strings.
Array is full.
John Lu Mary Leo


#-- Custom Test Cases --
```

End of Assignment.