Note: Deadline in 1/6/2023 evening

Task 1:

Create an empty vector of integers.

1. Ask the user to enter a series of integers and store them in the vector.
2. Print the contents of the vector.
3. Calculate and print the sum of all the integers in the vector.
4. Calculate and print the average of all the integers in the vector.
5. Find and print the largest and smallest values in the vector.

Task 2:

demonstrates exception handling with vectors in C++:

1. Create a vector of integers.
2. Ask the user to enter a series of integers and store them in the vector.
3. Implement error handling to catch the following exceptions:
    - If the read operation of the vector size exceeds the max vector limit.
4. Handle the exceptions by displaying the error message and allowing the user to re-enter the input or terminate the program gracefully.

Task 3:

implement a program that demonstrates the usage of static_cast, reinterpret_cast, and const_cast in C++:

1. Create a function called "convertValue" that takes an integer parameter.
2. Inside the function, use static_cast to convert the integer to a float and print the result.
3. Use reinterpret_cast to convert the integer to a pointer and print the address it points to.
4. Declare a constant integer variable and assign it a value.
5. Use const_cast to remove the const qualifier from the variable and modify its value.
6. Print the modified value of the variable.

Quiz:

Which of the following is a correct way to declare a vector of integers in C++?
a) vector<int> numbers;
b) int numbers[];
c) vector numbers;
d) int[] numbers;

What is the purpose of the push_back() function in C++ vectors?
a) It removes an element from the vector.
b) It adds an element at the beginning of the vector.
c) It adds an element at the end of the vector.
d) It sorts the elements in the vector.

How can you access the first element of a vector named "myVector" in C++?
a) myVector[0];
b) myVector.at(0);
c) myVector.front();
d) All of the above.

What is the function used to remove the last element from a vector in C++?
a) erase()
b) remove()
c) pop_back()
d) delete()

Which library is required to use vectors in C++?
a) iostream
b) vector
c) stdlib
d) vector.h

What is an exception in C++?
a) An error that occurs during runtime and disrupts the normal flow of the program.
b) A warning message displayed by the compiler.
c) A syntax error in the program.
d) An intentional termination of the program.

How do you catch an exception in C++?
a) Using the "try-catch" block.
b) Using the "if-else" statement.
c) Using the "switch-case" statement.
d) Using the "throw" keyword.

Which keyword is used to throw an exception in C++?
a) try
b) catch
c) throw
d) finally

What is the purpose of using exception handling in C++?
a) To gracefully handle errors and prevent program termination.
b) To optimize the program's performance.
c) To suppress error messages.
d) To increase the complexity of the program.

What is the purpose of static_cast in C++?
a) It performs safe type conversions between compatible types.
b) It converts a pointer to a different pointer type.
c) It converts a const-qualified object to a non-const object.
d) It performs low-level type conversions between unrelated types.

What is the purpose of reinterpret_cast in C++?
a) It performs safe type conversions between compatible types.
b) It converts a pointer to a different pointer type.
c) It converts a const-qualified object to a non-const object.
d) It performs low-level type conversions between unrelated types.

What is the purpose of const_cast in C++?
a) It performs safe type conversions between compatible types.
b) It converts a pointer to a different pointer type.
c) It converts a const-qualified object to a non-const object.
d) It performs low-level type conversions between unrelated types.

Which type of cast should be used to convert a pointer to a different pointer type without performing any checks?

a) static_cast
b) dynamic_cast
c) reinterpret_cast
d) const_cast

When should you use caution while using reinterpret_cast?
a) When converting between compatible types.
b) When converting between unrelated types.
c) When converting between const-qualified and non-const objects.
d) When converting between integer types.