

Part I: (True/False) - Write TRUE if the statement is correct and FALSE otherwise on the answer sheet ONLY (1 pt. each)

1. Both ofstream and fstream objects will create a file and then open the file if it does not exist.
2. All member functions of the class template are a template by default. However, the function definition outside the class must be a function template.
3. All types of pointers allocated a memory space from the stack at the compilation time only.
4. Unlike text files, storing data in a binary format is faster and can be easily transferred from one machine to another machine.
5. The name of the array refers to the memory address of the first element of the array and also to the entire array elements
6. By default, data/value is inserted and removed at the end of the vector. However, it is possible to insert or remove an element from the middle of a vector.
7. All STL containers can support all categories of iterators and algorithms.
8. Stream refers to a collection of data or a set of characters that may be a text or a program.
9. In dynamic memory allocation, the memory is acquired automatically from the heap and deallocated automatically as soon as the object goes out of the scope.

Part II: (Multiple Choice) - Choose the correct answer from the given alternatives and write the Letter of your choice on the answer sheet ONLY (1.5 pt. each)

1. Generic Programming is useful for, **except**:
 - a. To write an algorithm in term of types to be specified later
 - b. To develop algorithms with abstraction of data types
 - c. To make algorithms insensitive to the changes of data structure
 - d. To eliminate type casting/conversion and allow code reusability
 - e. None of the above
2. Identify **incorrect** statement(s) about template:
 - a. Template specialization is applicable in function template only.
 - b. Multiple template parameters are supported both in class and function templates.
 - c. Template parameter names cannot be used more than once in a template parameter list.
 - d. During the instantiation of the class template, the class name must be parameterized.
 - e. A and C

3. Given the vector declaration **vector <float> mark**, how to access the fourth element within the vector.
- a. mark[3] or mark.at(4)
 - b. mark[3] or mark.at(3)
 - c. mark[4] or mark.at(3)
 - d. mark[4] or mark.at(4)
 - e. A and C
4. One is **not** the component of STL.
- a. Containers
 - b. Algorithms
 - c. Class and Objects
 - d. Function templates
 - e. Generic smart pointers
5. What is/are the **limitations** of standard I/O streams?
- a. Temporary data storage
 - b. Limited storage capacity
 - c. Console is the only I/O data source
 - d. All of the above
 - e. A and C
 - f. A and B
6. How do smart pointers **overcome the drawback** of raw pointers?
- a. Avoid dangling pointer and memory leak
 - b. Allow both exclusive and shareable ownership of a resource
 - c. Auto management of dynamically allocated memory lifetime
 - d. A and B
 - e. All of the above
7. A weak smart pointer is described by;
- a. Similar to shared pointer, but cannot be dereferenced.
 - b. Cannot be deleted by other objects and it also affects the reference counter
 - c. Can be converted to a shared pointer using lock () function.
 - d. Used to avoid dangling pointer occurrence in shared pointer
 - e. C and D
 - f. All except B
8. Assume **ptr** is an integer pointer, what will happen when the statement **delete ptr**; is executed?
The memory space pointed by ptr;
- a. will be freed up and deallocated
 - b. Will be marked as ready to be reallocated
 - c. The content of ptr remains unchanged.
 - d. A and C
 - e. All of the above
-

9. A container class which is **not a complete container** by itself and that doesn't support an iterator.
- a. Adapter containers
 - b. Unordered associative containers
 - c. Sequence containers
 - d. Associative containers
 - e. C and D

10. A keyword(s) that is used in template declaration, **except**.

- a. typedef
- b. class
- c. typename
- d. template
- e. B and D

11. Assume the file contains 250 bytes of data. What is the output of the following code segment?

<pre>fstream f1; f1.open("data.txt", ios::out ios::ate); f1.seekp(-150, ios::end); f1.seekp(50, ios::curr); int ptr_loc = f1.getp(); cout<<ptr_loc;</pre>	<ul style="list-style-type: none">a. 100b. 150c. 200d. 50e. undefined
---	---

12. Which of the following gives **the [value] and the [address]** stored at the address pointed to by **the pointer ptr** respectively?

- a. ptr, *ptr
- b. &ptr, ptr
- c. *ptr, ptr
- d. ptr, &ptr
- e. *ptr, &ptr
- f. None

13. Given the statements, `int *p, i, k; i = 42; k = i; p = &i;`

Which of the following statements will change the value of **i** to 75?

- a. `k = 75;`
- b. `&k = 75;`
- c. `p = 75;`
- d. `*p = 75;`
- e. None

14. Consider the file opening statement in **Q11**, which of the following is the correct method to check for the success of file opening?

- a. `if(f1.is_open());`
- b. `if (f1 != NULL);`
- c. `if (f1);`
- d. A and B
- e. All are applicable

Part III: Debug and predict the output – 9 pts

Locate the error line, write the error and then determine the output assuming the correct code segment/program.

1. Given: code segments

1a) - 2.5 Pts	1b) – 2 Pts
<pre> 1. int intArray[5] = {50, 40, 30, 20, 10}; 2. int *pprint, **pprint; 3. pprint = intArray; 4. cout<< *(intArray + 2) << endl; 5. 6. pprint = & pprint; 7. pprint += 4; 8. cout<<(**pprint)++<<endl; 9. cout<<*(++pprint)<<endl; </pre>	<pre> 1. struct Fract { int nomi, deno; }; 2. int main() { 3. Fract f1={1, 2}; Fract f2 ={2, 5}; 4. Fract *result; 5. 6. result.nomi = (f1.nomi * f2.deno) 7. + (f2.nomi * f1.deno); 8. result.deno = f1.deno * f2.deno; 9. cout<<"Result="<<result.nomi<<"/" 10. cout<<result.deno<<endl; 11. } </pre>

2. Given: full program

2a) – 2 Pts	2b) - 2.5 Pts
<pre> 1. #include<iostream> 2. using namespace std; 3. 4. int main(){ 5. vector <int> mark; 6. 7. mark[0] = 19; 8. mark.insert(mark.begin(), 14); 9. mark.insert(mark.end()-1, 20); 10. mark.insert(mark.begin()+2, 13); 11. 12. auto vi = mark.begin(); 13. mark.erase(vi); 14. mark.pop_back(); 15. 16. cout<<"\nvector Size and capacity: "; 17. cout<<mark.size()<<mark.capacity(); 18. cout<<"\nFirst and Last elements: "; 19. cout<<mark.front()<<mark.back(); 20. } </pre>	<pre> 1. #include<iostream> 2. using namespace std; 3. 4. int main() { 5. unique_ptr<int> ptr1(new int); 6. shared_ptr<int> ptr2(new char); 7. *ptr1 = 100; *ptr2 = 'A'; 8. 9. unique_ptr<int> ptr3(new int); 10. shared_ptr<int> ptr4(new char); 11. ptr3 = ptr1; 12. ptr4 = ptr2 13. 14. cout<<ptr1<<endl; 15. cout<<*ptr1<<*ptr4<<endl; 16. cout<<ptr4.use_count()<<endl; 17. } </pre>

Part IV: (Write a program) – 11 Pts

1. Given below a 2D array declaration, write a code segment to print the array element using single loop with pointer. **(2 Pts)**

```
int product [][5] = {16, 22, 99, 4, 18, -78, 4, 11, 5, 98, 55, 6, 15, 2, 45, 33, 88, 72, 16, 3},
```

2. Mr. Fabec G. D. is a statistician and he is looking for a programmer who can design a program that reads data sets of various types (short int, long int, float, double etc.) and find the Mean (μ), variances (σ^2) and standard deviation (S). You are asked to design a program that performs the mentioned operation using the following program requirements. **(5 Pts)**
- The program should prompt the user to enter the size of the Data Set and the types.
 - Use a template function that accepts two parameters by reference and by value respectively. The 1st parameter is a vector whereas the 2nd parameter is the size of the data set.
 - The template function is used to read the data set, compute (mean, variance and standard deviation) and also print the data set elements and the computation results.
3. Consider the sample record below stored on a file named “vehicle.txt”, design a program that allows a user to modify the details of a vehicle.

Code	Model	Color	Year	Price
3462	BMW	Silver	2010	\$26434.62
6793	Ford	Black	2005	\$93765.83
7834	Lexus	White	2008	\$84376.89
9002	Nissan	Silver	2014	\$20904.75

The program should prompt the user to enter the **Code** of the vehicle in which its details will be updated and search within the record. If the **Code** is found, then the program allows the user to go through the vehicle details either to update or maintain the existing record of the vehicle. Finally, the modified record of the vehicle will be written back to the file. Verify that the file is opened successfully before performing the read/write operation. **(4 Pts)**