ZEESHAN INSTITUTE

VU ALL SERVICES AVAILABLE! 03135610637

HERE ARE THE SERVICES WE OFFER

- ✅COMPLETE/HALF LMS HANDLING
- ✅ASSIGNMENTS
- ✅QUIZZES
- ✅GDB
- ✅LECTURE WATCHING ASSISTANCE
- ✅ PAST PAPERS
- ✅HANDOUTS
- ✅PROJECTS
- ✅IMPORTANT FILES
- ✅CISCO ASSIGNMENT
- ✅IMPORTANT FILES AND RESOURCES
- ✅WITH 90-100 MARKS
- ✅GRAPHIC DESIGNING

🎯Get in touch now for complete exam preparation and boost your grades!

MUHAMMAD ZEESHAN

0313-5610637 0346-2027963

CS304 FINALTERM LATEST QUIZ SPRING 2026

Lecture 19: Stream Insertion & Extraction Operators

1. Which operator is used for stream insertion in C++? A) >> B) << C) <- D) ->

Correct Answer: B

2. Which operator is used for stream extraction? A) << B) >> C) != D) ==

Correct Answer: B

3. The object cout is an instance of which class? A) istream B) ostream C) iostream D) fstream

Correct Answer: B

4. The object cin is an instance of which class? A) ostream B) istream C) ifstream D) ios

Correct Answer: B

5. Why are stream insertion/extraction operators usually overloaded as friend functions? A) They must be private B) The left operand is not an object of the class C) They are too large to be member functions D) They cannot return references

Correct Answer: B

6. What is the return type of an overloaded stream insertion operator for chaining? A) void B) int C) ostream & D) istream &

Correct Answer: C

7. Where is the default output stream? A) File B) Network socket C) Console D) Printer

Correct Answer: C

8. Overloading stream operators allows us to treat user-defined types like: A) Functions B) Built-in types C) Constants D) Pointers

Correct Answer: B

Lecture 20: Subscript, Function, & Unary Operators

9. Which operator is the subscript operator? A) () B) [] C) {} D) .

Correct Answer: B

10. The subscript operator [] must be overloaded as: A) A friend function B) A static function C) A member function D) A global function

Correct Answer: C

11. Which operator is used to perform sub-string operations in some class designs? A) [] B) () C) -> D) *

Correct Answer: B

12. Unary operators operate on how many operands? A) Zero B) One C) Two D) Three

Correct Answer: B

13. Which of the following is a unary operator? A) + B) - (negation) C) * (multiplication) D) /

Correct Answer: B

14. When overloading a unary operator as a member function, how many arguments does it take? A) 0 B) 1 C) 2 D) None of these

Correct Answer: A

15. When overloading a unary operator as a friend function, how many arguments does it take? A) 0 B) 1 C) 2 D) 3

Correct Answer: B

16. Which of these operators cannot be overloaded? A) + B) :: C) [] D) ()

Correct Answer: B

Lecture 21: Increment/Decrement & Type Conversion

17. How does the compiler distinguish between prefix and postfix ++? A) Different return types B) A dummy int argument in postfix C) Using the virtual keyword D) Using different names

Correct Answer: B

18. What is the syntax for the prefix increment operator as a member function? A) operator++() B) operator++(int) C) ++operator() D) operator(int)++

Correct Answer: A

19. The prefix increment operator typically returns: A) A copy of the old value B) A reference to the updated object C) A constant integer D) Nothing

Correct Answer: B

20. The postfix increment operator typically returns: A) A reference to the object B) A copy of the object before incrementing C) A null pointer D) A boolean value

Correct Answer: B

21. Type conversion operators are used to: A) Change variable names B) Convert user-defined types to other types C) Delete objects D) Create new classes

Correct Answer: B

22. A type conversion operator has: A) Many arguments B) No return type specified in syntax C) A void return type D) A static keyword

Correct Answer: B

23. What is a drawback of type conversion operators? A) They are slow B) They can lead to unintended automatic conversions C) They require friend access D) They only work for int

Correct Answer: B

24. To avoid unintended conversions, programmers should use: A) Virtual functions B) Named conversion functions (e.g., AsInt()) C) Global variables D) Private constructors

Correct Answer: B

Lecture 22: Practical Implementation of Inheritance

25. Inheritance represents which type of relationship? A) HAS-A B) PART-OF C) IS-A D) USES

Correct Answer: C

26. In the declaration class Student : public Person, Person is the: A) Derived class B) Base class C) Child class D) Subclass

Correct Answer: B

27. What is the UML notation for inheritance? A) Solid line with a diamond B) Solid line with an arrow pointing to the base class C) Dotted line D) Line with a circle

Correct Answer: B

28. Which of the following is inherited by the derived class? A) Constructor B) Destructor C) Public member functions D) Assignment operator

Correct Answer: C

29. Does a derived class inherit private members of the base class? A) Yes, and it can access them directly B) Yes, but it cannot access them directly C) No D) Only if they are static

Correct Answer: B

30. Inheritance promotes: A) Redundancy B) Code Reuse C) Complexity D) Data Hiding only

Correct Answer: B

31. If Car is a Vehicle, which is the base class? A) Car B) Vehicle C) Both D) None

Correct Answer: B

32. Sub-typing is another name for: A) Encapsulation B) Extension/Inheritance C) Abstraction D) Composition

Correct Answer: B

Lecture 23: Access Specifiers in Inheritance

33. Which access specifier allows members to be accessible in derived classes but not in main()? A) Public B) Private C) Protected D) Global

Correct Answer: C

34. Under public inheritance, public members of the base class become _____ in the derived class. A) Public B) Private C) Protected D) Static

Correct Answer: A

35. Under public inheritance, protected members of the base class become _____ in the derived class. A) Public B) Private C) Protected D) Internal

Correct Answer: C

36. Private members of the base class are: A) Accessible in derived class B) Inaccessible in derived class C) Accessible in main D) Accessible through pointers only

Correct Answer: B

37. The static type of a pointer is: A) The type of the object it points to at runtime B) The type specified at declaration C) Always void * D) Determined by the compiler at runtime

Correct Answer: B

38. If a Person *ptr points to a Student object, its static type is: A) Student B) Person C) Object D) Pointer

Correct Answer: B

39. The IS-A relationship should be: A) One-way B) Bi-directional C) Random D) Static

Correct Answer: A

40. Protected access is a compromise between: A) Static and Dynamic B) Public and Private C) Class and Object D) Inheritance and Composition

Correct Answer: B

Lecture 24: Constructors in Inheritance

41. In inheritance, which constructor is called first? A) Derived class constructor B) Base class constructor C) Both at the same time D) Destructor

Correct Answer: B

42. In inheritance, which destructor is called first? A) Base class destructor B) Derived class destructor C) Destructors are not called D) Random order

Correct Answer: B

43. To call a parameterized base constructor, we use: A) The dot operator B) Member Initializer List C) The this pointer D) A friend function

Correct Answer: B

44. If the base class has no default constructor, the derived class: A) Can use its own default B) Must explicitly call a base constructor C) Will not compile D) Will crash at runtime

Correct Answer: B

45. Constructor calls follow a: A) Top-to-bottom order in hierarchy B) Bottom-to-top order C) Random order D) Left-to-right order

Correct Answer: A

46. Destructor calls follow a: A) Top-to-bottom order B) Bottom-to-top order C) Reverse order of constructors D) Both B and C

Correct Answer: D

47. A derived class object contains: A) Only its own members B) Only base members C) Both base and its own members D) Neither

Correct Answer: C

48. A modified default constructor in inheritance refers to: A) A constructor with no arguments B) A constructor calling a base constructor C) A virtual constructor D) A static constructor

Correct Answer: B

Lecture 25: Overloading vs. Overriding

49. Redefining a base class function in a derived class with the same signature is called: A) Overloading B) Overriding C) Encapsulation D) Virtualization

Correct Answer: B

50. Defining multiple functions with the same name but different signatures in the same class is: A) Overloading B) Overriding C) Abstraction D) Composition

Correct Answer: A

51. Overriding occurs across: A) Different classes in inheritance B) The same class C) Different programs D) Global scope

Correct Answer: A

52. To call an overridden base class function from the derived class, use: A) this->function() B) Base::function() C) super.function() D) function()

Correct Answer: B

53. Overriding requires the functions to have the same: A) Name only B) Signature and name C) Return type only D) Number of arguments only

Correct Answer: B

54. A class hierarchy is: A) A list of variables B) A tree-like structure of related classes C) A single class with many objects D) A set of friend functions

Correct Answer: B

55. If a function is overridden, the derived class object: A) Cannot access the base version B) Accesses the derived version by default C) Always calls both versions D) Crashes the program

Correct Answer: B

56. Overloading is resolved at: A) Runtime B) Compile time C) Link time D) Design time

Correct Answer: B

Lecture 26: Base Initialization & Private Inheritance

57. In private inheritance, public members of the base class become _____ in the derived class. A) Public B) Private C) Protected D) Hidden

Correct Answer: B

58. In private inheritance, protected members of the base class become _____ in the derived class. A) Public B) Private C) Protected D) Global

Correct Answer: B

59. Private inheritance is often used to implement: A) IS-A relationship B) "Implemented in terms of" relationship C) PART-OF relationship D) Global access

Correct Answer: B

60. Is an object of a privately inherited class compatible with a base class pointer? A) Yes B) No C) Only if casted D) Only in main

Correct Answer: B

61. What is the default inheritance for a class? A) Public B) Private C) Protected D) Static

Correct Answer: B

62. What is the default inheritance for a struct? A) Public B) Private C) Protected D) Friend

Correct Answer: A

63. Base initialization occurs in the: A) Constructor body B) Member Initializer List C) Destructor D) main() function

Correct Answer: B

64. Private inheritance restricts: A) Code reuse B) External access to base members C) Memory usage D) Compilation speed

Correct Answer: B

Lecture 27: Specialization & Protected Inheritance

65. Restriction (Specialization) in inheritance means: A) Adding new features B) Limiting the behavior of the base class C) Inheriting everything D) Deleting the base class

Correct Answer: B

66. In protected inheritance, public members of the base class become _____ in the derived class. A) Public B) Private C) Protected D) Static

Correct Answer: C

67. In protected inheritance, protected members of the base class become _____ in the derived class. A) Public B) Private C) Protected D) Global

Correct Answer: C

68. Protected inheritance allows: A) Access from main() B) Further derivation with access to protected members C) Universal access D) No access at all

Correct Answer: B

69. A class hierarchy using specialization is harder to: A) Write B) Maintain C) Understand D) None of these

Correct Answer: B

70. Protected inheritance is less common than: A) Private inheritance B) Public inheritance C) Composition D) Global variables

Correct Answer: B

71. Members of a base class that should be hidden from everyone except children should be: A) Public B) Private C) Protected D) Static

Correct Answer: C

72. Can a base class pointer point to a protectedly derived class object? A) Yes B) No (outside the class/children) C) Only if static D) Always

Correct Answer: B

Lecture 28: Virtual Functions

73. A virtual function is declared using the keyword: A) static B) virtual C) inline D) friend

Correct Answer: B

74. Virtual functions enable: A) Static binding B) Dynamic binding C) Compile-time polymorphism D) Data hiding

Correct Answer: B

75. Dynamic binding is also known as: A) Late binding B) Early binding C) Static dispatch D) Fast binding

Correct Answer: A

76. To achieve polymorphic behavior, we must use: A) Objects B) Pointers or references C) Static variables D) Friend functions

Correct Answer: B

77. Static binding is performed by the: A) Runtime system B) Compiler C) User D) Operating system

Correct Answer: B

78. Which of the following can be virtual? A) Constructor B) Destructor C) Static function D) Global function

Correct Answer: B

79. A class with a virtual function: A) Is always abstract B) Is larger due to V-table pointer C) Cannot be inherited D) Has no constructor

Correct Answer: B

80. Polymorphism means: A) One name, one form B) One name, many forms C) Many names, one form D) No name

Correct Answer: B

Lecture 29: Abstract Classes

81. A pure virtual function is declared by: A) virtual void func() = 0; B) void func() = virtual; C) virtual void func() {} D) pure virtual void func();

Correct Answer: A

82. An abstract class is a class that: A) Has no data members B) Has at least one pure virtual function C) Cannot be used as a base class D) Has all static members

Correct Answer: B

83. Can we instantiate an object of an abstract class? A) Yes B) No C) Only if it has a constructor D) Only in main()

Correct Answer: B

84. A class derived from an abstract class that implements all pure virtual functions is: A) Another abstract class B) A concrete class C) A static class D) A virtual class

Correct Answer: B

85. What is a V-table? A) A table of variables B) A table of virtual function addresses C) A table of class names D) A table of objects

Correct Answer: B

86. Every object of a class with virtual functions contains a: A) V-table B) VPTR (Virtual Pointer) C) Static pointer D) Copy of the class

Correct Answer: B

87. V-tables are created: A) Per object B) Per class C) Per function D) Per program

Correct Answer: B

88. Pure virtual functions are used to: A) Make the program faster B) Define an interface for derived classes C) Hide data D) Prevent inheritance

Correct Answer: B

Lecture 30: Polymorphism Case Study

89. In a Shape hierarchy, why is Shape often an abstract class? A) It is too generic to exist as an object B) It has no color C) It is static D) It is private

Correct Answer: A

90. Why should you never use arrays polymorphically? A) Pointers are 8 bytes B) Array indexing uses the static type's size C) Arrays cannot hold pointers D) It is too fast

Correct Answer: B

91. If a base class pointer array holds derived class objects, incrementing the pointer: A) Always moves to the next derived object B) Moves by the size of the base class object C) Moves by 1 byte D) Crashes immediately

Correct Answer: B

92. The size of a class with virtual functions is increased by: A) The size of the V-table B) The size of a pointer (VPTR) C) 100 bytes D) Nothing

Correct Answer: B

93. Polymorphism helps in: A) Code maintenance B) Reducing switch-case logic C) Extensibility D) All of the above

Correct Answer: D

94. The getType() function is often used to avoid: A) Virtual functions B) Static variables C) Switch-case blocks D) Recursion

Correct Answer: C

95. A Payroll system uses polymorphism to: A) Calculate salary for different types of employees B) Print names only C) Save memory D) Hide the employee class

Correct Answer: A

96. "Dynamic dispatch" refers to: A) Compile-time call resolution B) Runtime call resolution C) Loading a DLL D) Deleting an object

Correct Answer: B

Lecture 31: Multiple Inheritance

97. Multiple inheritance is when a class inherits from: A) One base class B) More than one base class C) A virtual class D) Itself

Correct Answer: B

98. The syntax for multiple inheritance is: A) class D : B1, B2 B) class D : public B1, public B2 C) class D : B1 : B2 D) class D (B1, B2)

Correct Answer: B

99. The "Diamond Problem" occurs in: A) Single inheritance B) Multiple inheritance C) Composition D) Abstraction

Correct Answer: B

100. How is the diamond problem solved in C++? A) Multiple inheritance B) Virtual inheritance C) Static inheritance D) Private inheritance

Correct Answer: B

101. Virtual inheritance ensures: A) The base class is private B) Only one instance of the shared base class exists C) The program runs faster D) No constructors are called

Correct Answer: B

102. To use virtual inheritance, use the keyword: A) shared B) virtual C) base D) friend

Correct Answer: B

103. Ambiguity in multiple inheritance happens when: A) Classes have different names B) Base classes have functions with the same name C) There are no constructors D) Memory is low

Correct Answer: B

104. Multiple inheritance can lead to: A) Simple code B) Complexity and ambiguity C) Faster compilation D) Better encapsulation

Correct Answer: B

Lecture 32: Generic Programming & Templates

105. Generic programming aims to: A) Use only integers B) Write code independent of data types C) Make code specific to one class D) Use global variables

Correct Answer: B

106. In C++, generic programming is achieved through: A) Inheritance B) Templates C) Macros D) Friend functions

Correct Answer: B

107. A function template is defined using: A) template <class T> B) function <T> C) generic <T> D) void template T

Correct Answer: A

108. The T in template <class T> is called a: A) Type parameter B) Type constant C) Type variable D) Template object

Correct Answer: A

109. Template code is generated by the compiler during: A) Linking B) Instantiation C) Execution D) Pre-processing

Correct Answer: B

110. Can a template function handle user-defined types? A) No B) Yes, if they support required operators C) Only if they are static D) Only if they are pointers

Correct Answer: B

111. The keyword typename can be used instead of: A) struct B) class C) int D) void

Correct Answer: B

112. Templates help in: A) Avoiding code duplication for different types B) Faster runtime C) Hiding code D) Manual memory management

Correct Answer: A

Lecture 33: Multiple Type Arguments & Policies

113. Can a template have multiple parameters? A) No B) Yes, separated by commas C) Only two D) Only if they are the same type

Correct Answer: B

114. Template arguments can be used as: A) Policies B) Global variables C) Destructors D) Comments

Correct Answer: A

115. A policy in templates is: A) A set of rules for the compiler B) A strategy passed as a template parameter C) A private member D) A type of inheritance

Correct Answer: B

116. Overloading vs. Templates: Templates are preferred when: A) Behavior is different for each type B) Behavior is the same for all types C) No functions are needed D) Types are unknown

Correct Answer: B

117. Can a template have default values for parameters? A) No B) Yes (for class templates) C) Only for int D) Only in C

Correct Answer: B

118. Which solution uses a default policy? A) First B) Second C) Third D) All of them

Correct Answer: C

119. Template arguments as policy allow: A) Flexibility in class behavior B) Less code C) Static types D) Private access

Correct Answer: A

120. User-defined types in templates: A) Must be explicitly declared B) Must overload operators used by the template C) Cannot be used D) Must be virtual

Correct Answer: B

Lecture 34: Generic Algorithms & Class Templates

121. A generic algorithm is: A) A math function B) An algorithm that works on any container C) A sorting algorithm for int only D) A recursive function

Correct Answer: B

122. A class template is defined as: A) template <class T> class MyClass { ... }; B) class <T> MyClass { ... }; C) template MyClass { ... }; D) class MyClass template <T> { ... };

Correct Answer: A

123. To instantiate a class template Vector for int, use: A) Vector v; B) Vector<int> v; C) Vector(int) v; D) int Vector v;

Correct Answer: B

124. Member functions of a template class defined outside the class require: A) No special syntax B) template <class T> prefix C) static keyword D) friend keyword

Correct Answer: B

125. Class templates allow: A) Creating generic data structures like Linked Lists B) Faster execution C) Private inheritance only D) Global scope

Correct Answer: A

126. Generic algorithms often use: A) Hardcoded types B) Template parameters for types C) Void pointers D) Global arrays

Correct Answer: B

127. Class templates increase: A) Runtime speed B) Reusability C) Source code size D) Memory leaks

Correct Answer: B

128. Can a class template have non-type parameters (e.g., int size)? A) No B) Yes C) Only in constructors D) Only in main

Correct Answer: B

Lecture 35: Member Templates & Specialization

129. Template specialization is used when: A) A specific type requires different logic B) The program is too slow C) We want to hide the template D) Inheritance fails

Correct Answer: A

130. A specialized template for char * is often needed because: A) char * is not a type B) Normal comparison/assignment doesn't work for C-strings C) It is too large D) It is private

Correct Answer: B

131. The syntax for a complete specialization starts with: A) template <class T> B) template <> C) special template D) template <void>

Correct Answer: B

132. Member templates are: A) Templates inside a class B) Templates for data members C) Static templates D) Global templates

Correct Answer: A

133. Specialization helps in: A) Optimizing for specific types B) Handling pointers correctly C) Both A and B D) None

Correct Answer: C

134. If both a general template and a specialization match, which is used? A) General B) Specialization C) Random D) Compiler error

Correct Answer: B

135. Can you specialize only some functions of a class template? A) No B) Yes C) Only the constructor D) Only if they are static

Correct Answer: B

136. Specialization allows for: A) Efficient code for specific types B) Generic code C) Hiding types D) Deleting types

Correct Answer: A

Lecture 36: Partial Specialization & Non-type Parameters

137. Partial specialization is: A) Specializing for some but not all template parameters B) Writing half a function C) Specializing only for int D) Not possible in C++

Correct Answer: A

138. Can function templates be partially specialized? A) Yes B) No (only class templates) C) Only in C++11 D) Only for float

Correct Answer: B

139. Complete specialization is when: A) All parameters are fixed to specific types B) One parameter is fixed C) No parameters are fixed D) The template is deleted

Correct Answer: A

140. Non-type parameters must be: A) Integers or pointers B) Floats C) Strings D) Objects

Correct Answer: A

141. A default non-type parameter looks like: A) template <int size = 10> B) template <int size> C) template <size = 10> D) template <class T = 10>

Correct Answer: A

142. Default type parameters look like: A) template <class T = int> B) template <int T = class> C) template <T = int> D) template <class T>

Correct Answer: A

143. Using different specializations: A) Increases binary size B) Decreases source code C) Both A and B D) Neither

Correct Answer: A

144. Partial specialization is useful for: A) Handling pointer types T* B) Handling only int C) Hiding logic D) Speed

Correct Answer: A

Lecture 37: Resolution Order & Inheritance

145. What is the resolution order for a function call? A) Ordinary function > Specialized template > General template B) General template > Ordinary function C) Random D) Smallest function first

Correct Answer: A

146. A template class can inherit from: A) A non-template class B) Another template class C) Both A and B D) None

Correct Answer: C

147. A non-template class can inherit from: A) A specific instance of a template class B) A general template C) A friend function D) A static member

Correct Answer: A

148. Inheritance with templates requires: A) Careful syntax for base class calls B) No special changes C) Static members D) Virtual functions

Correct Answer: A

149. Is a template class instance a real class? A) No B) Yes, once instantiated C) Only at runtime D) Only if it's public

Correct Answer: B

150. Can a template class have virtual functions? A) No B) Yes C) Only if they are static D) Only in main

Correct Answer: B

151. Derivation from a template class: A) Follows standard inheritance rules B) Is not allowed C) Requires the friend keyword D) Is always private

Correct Answer: A

152. Resolution order is determined by the: A) Linker B) Compiler C) Programmer D) OS

Correct Answer: B

Lecture 38: Templates & Friends

153. A non-template friend function of a template class is a friend to: A) All instances of the template B) No instances C) Only one instance D) Only the static members

Correct Answer: A

154. Rule 1 of templates and friends involves: A) Non-template friend function B) Template friend function C) Virtual friends D) Static friends

Correct Answer: A

155. Rule 2 involves a friend function: A) That is a template itself B) That is a member of another class C) That is private D) That is static

Correct Answer: A

156. Rule 3 states that a specific instance of a template can be a: A) Friend B) Constructor C) Destructor D) Variable

Correct Answer: A

157. Can a class template have a friend class? A) No B) Yes C) Only if it's a template D) Only in C

Correct Answer: B

158. Rule 4 involves: A) A friend template class B) A friend static function C) A friend global variable D) A friend constructor

Correct Answer: A

159. Friends and templates can lead to: A) Tight coupling B) Better performance C) Security issues D) Simpler code

Correct Answer: A

160. Which rule says a friend can be a specific instance like Vector<int>? A) Rule 1 B) Rule 2 C) Rule 3 D) Rule 4

Correct Answer: C

Lecture 39: Templates Conclusion & Problems

161. Static members in a template class are: A) Shared by all instances (int, float, etc.) B) Unique for each specific instance (e.g., Vector<int> has its own) C) Global D) Not allowed

Correct Answer: B

162. A problem with basic generic algorithms is they: A) Might not support multiple traversals B) Are too generic C) Only work for integers D) Use too much RAM

Correct Answer: A

163. Inconsistent behavior in algorithms occurs when: A) Different containers return different markers B) Code is written in C C) Types are small D) Errors are caught

Correct Answer: A

164. Using a whole container as a marker is: A) Efficient B) Inefficient and inconsistent C) Standard practice D) Virtual

Correct Answer: B

165. Traversal strategy refers to: A) How we move through data B) How we delete data C) How we name variables D) How we compile

Correct Answer: A

166. Generic algorithms are powerful because they promote: A) Reuse B) Specificity C) Complexity D) Speed only

Correct Answer: A

167. Does Vector<int> and Vector<double> share the same static variable count? A) Yes B) No C) Only if public D) Only if virtual

Correct Answer: B

168. To solve traversal problems, we use: A) Iterators B) Static variables C) Global arrays D) More templates

Correct Answer: A

Lecture 40: Cursors & Iterators

169. A cursor is: A) A mouse pointer B) A simple marker for a position in a container C) A complex object D) A virtual function

Correct Answer: B

170. An iterator is an object that: A) Emulates a pointer to traverse a container B) Deletes a container C) Is a base class D) Is always static

Correct Answer: A

171. Iterators contribute to: A) Data abstraction B) Speed C) Memory usage D) Global scope

Correct Answer: A

172. Advantages of iterators include: A) Multiple traversals on one container B) Changing traversal strategy without changing the container C) Emulating pointers D) All of the above

Correct Answer: D

173. A container should support which operations for generic algorithms? A) start(), beyond(), next() B) add(), remove() C) print(), sort() D) static, virtual

Correct Answer: A

174. Iterators are: A) Classes themselves B) Integers C) Strings D) Macros

Correct Answer: A

175. The * operator for an iterator: A) Multiplies B) Dereferences to get the value C) Deletes D) Moves to the next

Correct Answer: B

176. The ++ operator for an iterator: A) Increments the value B) Moves to the next element C) Resets D) Stops

Correct Answer: B

Lecture 41: Standard Template Library (STL)

177. STL stands for: A) Standard Type Library B) Standard Template Library C) Static Template Library D) Standard Tool Library

Correct Answer: B

178. The three key components of STL are: A) Containers, Iterators, Algorithms B) Classes, Objects, Functions C) Files, Streams, Buffers D) Vectors, Lists, Queues

Correct Answer: A

179. STL promotes reuse by: A) Providing pre-written, tested code B) Hiding code C) Using global variables D) Forcing inheritance

Correct Answer: A

180. A sequence container organizes elements in a: A) Tree B) Linear arrangement C) Hash table D) Random order

Correct Answer: B

181. Which STL container allows random access? A) list B) vector C) map D) set

Correct Answer: B

182. A deque is a: A) Double-ended queue B) Single-ended queue C) Stack D) Linked list

Correct Answer: A

183. Which container is implemented as a doubly linked list? A) vector B) deque C) list D) stack

Correct Answer: C

184. The function push_back(): A) Adds an element at the end B) Adds an element at the front C) Removes an element D) Sorts the container

Correct Answer: A

Lecture 42: Iterator Categories & Algorithms

185. Which iterator category is the most powerful? A) Input B) Output C) Forward D) Random Access

Correct Answer: D

186. Bidirectional iterators support: A) Only ++ B) Both ++ and -- C) Only [] D) Only *

Correct Answer: B

187. Forward iterators support: A) ++ only B) -- only C) Random access D) All math

Correct Answer: A

188. Container adapters include: A) stack, queue, priority_queue B) vector, list C) map, set D) int, float

Correct Answer: A

189. STL algorithms work with: A) Iterators B) Pointers C) Both A and B D) None

Correct Answer: C

190. std::find() is an example of an STL: A) Container B) Iterator C) Algorithm D) Class

Correct Answer: C

191. Associative containers include: A) vector B) map, set C) stack D) deque

Correct Answer: B

192. Random access iterators allow: A) it + n B) it - n C) it[n] D) All of the above

Correct Answer: D

Lecture 43: Exception Handling

193. Abnormal termination happens when: A) The program finishes normally B) An error occurs and the program crashes C) The user clicks exit D) A file is saved

Correct Answer: B

194. Graceful termination means: A) Crashing with a blue screen B) Handling errors and exiting properly C) Deleting all files D) Restarting the PC

Correct Answer: B

195. The keyword to start a block of code to be monitored for errors is: A) catch B) try C) throw D) finally

Correct Answer: B

196. The keyword used to signal that an error has occurred is: A) signal B) error C) throw D) alert

Correct Answer: C

197. The keyword to handle a thrown exception is: A) handle B) receive C) catch D) trap

Correct Answer: C

198. An exception is: A) A normal part of code B) An unusual condition or error C) A variable D) A comment

Correct Answer: B

199. Can we throw any type of object (int, string, etc.)? A) No, only integers B) Yes C) Only classes D) Only pointers

Correct Answer: B

200. If no catch block matches the thrown exception: A) The program continues B) The program terminates (calls terminate()) C) It is ignored D) The computer restarts

Correct Answer: B

Lecture 44: Stack Unwinding

201. Stack unwinding is the process of: A) Creating new objects B) Destroying local objects in the path of a thrown exception C) Clearing the RAM D) Reversing a linked list

Correct Answer: B

202. During stack unwinding, which functions are called? A) Constructors B) Destructors C) Static functions D) Main

Correct Answer: B

203. Exception handling process starts with: A) catch B) throw C) new D) delete

Correct Answer: B

204. Can a catch block re-throw an exception? A) No B) Yes, using throw; C) Only once D) Only in Java

Correct Answer: B

205. A catch-all block is written as: A) catch(...) B) catch(all) C) catch(*) D) catch(void)

Correct Answer: A

206. Stack unwinding continues until: A) A matching catch is found B) The program ends C) main() is reached D) Both A and C

Correct Answer: D

207. Exceptions help in separating: A) Classes and objects B) Error-handling code from normal logic C) Data and functions D) Static and dynamic members

Correct Answer: B

208. Is terminate() called if an exception occurs in a destructor during unwinding? A) Yes B) No C) Sometimes D) Only if caught

Correct Answer: A

Lecture 45: Resource Management

209. Resource management in exceptions is critical because: A) Exceptions can cause memory leaks if delete is skipped B) RAM is infinite C) Files close automatically D) It is optional

Correct Answer: A

210. A solution for resource management is: A) Using global variables B) RAII (Resource Acquisition Is Initialization) C) Not using exceptions D) Closing files manually every time

Correct Answer: B

211. If an exception is thrown after new but before delete, the memory is: A) Freed B) Leaked C) Reused D) Virtualized

Correct Answer: B

212. Destructors are essential for: A) Resource cleanup B) Creating objects C) Printing data D) Naming classes

Correct Answer: A

213. Exception handling provides a way to: A) Ignore errors B) Transfer control from point of error to a handler C) Speed up code D) Encapsulate data

Correct Answer: B

214. The exception handling mechanism is: A) Synchronous B) Asynchronous C) Random D) Parallel

Correct Answer: A

215. Resource management includes: A) Memory B) Files C) Sockets D) All of the above

Correct Answer: D

216. "Stack Unwinding" is a feature of: A) C B) C++ Exception Handling C) Assembly D) HTML

Correct Answer: B