C++ VIVA Questions

1. What is the full form of OOPS?

   Object Oriented Programming System.

2. What is a class?

   Class is a blue print which reflects the entities attributes and actions. Technically defining a class is designing an user defined data type.

3. What is an object?

   An instance of the class is called as object.

4. List the types of inheritance supported in C++.

   Single, Multilevel, Multiple, Hierarchical and Hybrid.

5. What is the role of protected access specifier?

   If a class member is protected then it is accessible in the inherited class. However, outside the both the private and protected members are not accessible.

6. What is encapsulation?

   The process of binding the data and the functions acting on the data together in an entity (class) called as encapsulation.

7. What is abstraction?

   Abstraction refers to hiding the internal implementation and exhibiting only the necessary details.

8. What is inheritance?

   Inheritance is the process of acquiring the properties of the existing class into the new class. The existing class is called as base/parent class and the inherited class is called as derived/child class.

9. Explain the purpose of the keyword volatile.

   Declaring a variable volatile directs the compiler that the variable can be changed externally. Hence avoiding compiler optimization on the variable reference.

10. What is an inline function?

A function prefixed with the keyword inline before the function definition is called as inline function. The inline functions are faster in execution when compared to normal functions as the compiler treats inline functions as macros.

11. What is a storage class?

Storage class specifies the life or scope of symbols such as variable or functions.

12. Mention the storage classes names in C++.

The following are storage classes supported in C++

auto, static, extern, register and mutable

13. What is the role of mutable storage class specifier?

A constant class object's member variable can be altered by declaring it using mutable storage class specifier. Applicable only for non-static and non-constant member variable of the class.

14. Distinguish between shallow copy and deep copy.

Shallow copy does memory dumping bit-by-bit from one object to another. Deep copy is copy field by field from object to another. Deep copy is achieved using copy constructor and or overloading assignment operator.

15. What is a pure virtual function?

A virtual function with no function body and assigned with a value zero is called as pure virtual function.

16. What is an abstract class in C++?

A class with at least one pure virtual function is called as abstract class. We cannot instantiate an abstract class.

17. What is a reference variable in C++?

A reference variable is an alias name for the existing variable. Which mean both the variable name and reference variable point to the same memory location. Therefore updation on the original variable can be achieved using reference variable too.

18. What is role of static keyword on class member variable?

A static variable does exit though the objects for the respective class are not created. Static member variable share a common memory across all the objects created for the respective class. A static member variable can be referred using the class name itself.

19. Explain the static member function.

A static member function can be invoked using the class name as it exits before class objects comes into existence. It can access only static members of the class.

20. Name the data type which can be used to store wide characters in C++.

wchar_t

21. What are/is the operator/operators used to access the class members?

Dot (.) and Arrow (->)

22. Can we initialize a class/structure member variable as soon as the same is defined?

No, Defining a class/structure is just a type definition and will not allocated memory for the same.

23. What is the data type to store the Boolean value?

bool, is the new primitive data type introduced in C++ language.

24. What is function overloading?

Defining several functions with the same name with unique list of parameters is called as function overloading.

25. What is operator overloading?

Defining a new job for the existing operator w.r.t the class objects is called as operator overloading.

26. Do we have a String primitive data type in C++?

No, it's a class from STL (Standard template library).

27. Name the default standard streams in C++.

    cin, cout, cerr and clog.

28. Which access specifier/s can help to achive data hiding in C++?

    Private & Protected.

29. When a class member is defined outside the class, which operator can be used to associate the function definition to a particular class?

    Scope resolution operator (::)

30. What is a destructor? Can it be overloaded?

    A destructor is the member function of the class which is having the same name as the class name and prefixed with tilde (~) symbol. It gets executed automatically w.r.t the object as soon as the object loses its scope. It cannot be overloaded and the only form is without the parameters.

31. What is a constructor?

    A constructor is the member function of the class which is having the same as the class name and gets executed automatically as soon as the object for the respective class is created.

32. What is a default constructor? Can we provide one for our class?

    Every class does have a constructor provided by the compiler if the programmer doesn't provides one and known as default constructor. A programmer provided constructor with no parameters is called as default constructor. In such case compiler doesn't provides the constructor.

33. Which operator can be used in C++ to allocate dynamic memory?

    'new' is the operator can be used for the same.

34. What is the purpose of 'delete' operator?

    'delete' operator is used to release the dynamic memory which was created using 'new' operator.

35. Can I use malloc() function of C language to allocate dynamic memory in C++?

    Yes, as C is the subset of C++, we can all the functions of C in C++ too.

36. Can I use ‘delete’ operator to release the memory which was allocated using malloc() function of C language?

    No, we need to use free() of C language for the same.

37. What is a friend function?

    A function which is not a member of the class but still can access all the member of the class is called so. To make it happen we need to declare within the required class following the keyword ‘friend’.

38. What is a copy constructor?

    A copy constructor is the constructor which take same class object reference as the parameter. It gets automatically invoked as soon as the object is initialized with another object of the same class at the time of its creation.

39. Does C++ supports exception handling? If so what are the keywords involved in achieving the same.

    C++ does supports exception handling. try, catch & throw are keyword used for the same.

40. Explain the pointer – this.

    This, is the pointer variable of the compiler which always holds the current active object’s address.

41. What is the difference between the keywords struct and class in C++?

    By default the members of struct are public and by default the members of the class are private.

42. Can we implement all the concepts of OOPS using the keyword struct?

    Yes.

43. What is the block scope variable in C++?

A variable whose scope is applicable only within a block is said so. Also a variable in C++ can be declared anywhere within the block.

44. What is the role of the file opening mode ios::trunk?

    If the file already exists, its content will be truncated before opening the file.

45. What is the scope resolution operator?

    The scope resolution operator is used to Resolve the scope of global variables.

    To associate function definition to a class if the function is defined outside the class.

46. What is a namespace?

    A namespace is the logical division of the code which can be used to resolve the name conflict of the identifiers by placing them under different name space.

47. What are command line arguments?

    The arguments/parameters which are sent to the main() function while executing from the command line/console are called so. All the arguments sent are the strings only.

48. What is a class template?

    A template class is a generic class. The keyword template can be used to define a class template.

49. How can we catch all kind of exceptions in a single catch block?

    The catch block with ellipses as follows

```
catch(…)

{

}
```

50. What is keyword auto for?

    By default every local variable of the function is automatic (auto). In the below function both the variables ‘i’ and ‘j’ are automatic variables.

```
void f()
{
int i;
auto int j;
}
```

NOTE: A global variable can't be an automatic variable.

51. What is a static variable?

    A static local variables retains its value between the function call and the default value is 0. The following function will print 1 2 3 if called thrice.

```
void f()
{
static int i;
++i;
cout<<i;
}
```

If a global variable is static then its visibility is limited to the same source code.

52. What is the purpose of extern storage specifier.

    Used to resolve the scope of global symbol

    #include <iostream>

    using namespace std;

    main()

    {

    extern int i;

    cout<<i<<endl;

```
}
int i=20;
```

53. What is the meaning of base address of the array?

    The starting address of the array is called as the base address of the array.

54. When should we use the register storage specifier?

    If a variable is used most frequently then it should be declared using register storage specifier, then possibly the compiler gives CPU register for its storage to speed up the look up of the variable.

55. Can a program be compiled without main() function?

    Yes, it can be but cannot be executed, as the execution requires main() function definition.

56. Where an automatic variable is stored?

    Every local variable by default being an auto variable is stored in stack memory

57. What is a container class?

    A class containing at least one member variable of another class type in it is called so.

58. What is a token?

    A C++ program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.

59. What is a preprocessor?

    Preprocessor is a directive to the compiler to perform certain things before the actual compilation process begins.

60. What are command line arguments?

    The arguments which we pass to the main() function while executing the program are called as command line arguments. The parameters are always strings held in the

second argument (below in args) of the function which is array of character pointers. First argument represents the count of arguments (below in count) and updated automatically by operating system.

```
main( int count, char *args[]) {

}
```

61. What are the different ways of passing parameters to the functions? Which to use when?

    Call by value: We send only values to the function as parameters. We choose this if we do not want the actual parameters to be modified with formal parameters but just used.

    Call by address: We send address of the actual parameters instead of values. We choose this if we do want the actual parameters to be modified with formal parameters.

    Call by reference: The actual parameters are received with the C++ new reference variables as formal parameters. We choose this if we do want the actual parameters to be modified with formal parameters.

62. What is reminder for 5.0 % 2?

    Error, It is invalid that either of the operands for the modulus operator (%) is a real number.

63. Can we resize the allocated memory which was allocated using ‘new’ operator?

    No, there is no such provision available.

64. Who designed C++ programming language?

    Bjarne Stroustrup.

65. Which operator can be used to determine the size of a data type/class or variable/object?

    sizeof

66. How can we refer to the global variable if the local and the global variable names are same?

We can apply scope resolution operator (::) to the for the scope of global variable.

67. What are valid operations on pointers?

The only two permitted operations on pointers are

Comparision ii) Addition/Substraction (excluding void pointers)

68. What is recursion?

Function calling itself is called as recursion.

69. What is the maximum length of an identifier?

Ideally it is 32 characters and also implementation dependent.

70. What is the default function call method?

By default the functions are called by value.

71. What are available mode of inheritance to inherit one class from another?

Public, private & protected

72. What is the difference between delete and delete[]?

Delete[] is used to release the array allocated memory which was allocated using new[] and delete is used to release one chunk of memory which was allocated using new.

73. Does an abstract class in C++ need to hold all pure virtual functions?

Not necessarily, a class having at least one pure virtual function is abstract class too.

74. Is it legal to assign a base class object to a derived class pointer?

No, it will be error as the compiler fails to do conversion.

75. What happens if an exception is thrown outside a try block?

The program shall quit abruptly.

76. Are the exceptions and error same?

No, exceptions can be handled whereas program cannot resolve errors.

77. What is function overriding?

    Defining the functions within the base and derived class with the same signature and name where the base class's function is virtual.

78. Which function is used to move the stream pointer for the purpose of reading data from stream?

    seekg()

79. Which function is used to move the stream pointer for the purpose of writing data from stream?

    seekp()

80. Are class functions taken into consideration as part of the object size?

    No, only the class member variables determines the size of the respective class object.

81. Can we create and empty class? If so what would be the size of such object.

    We can create an empty class and the object size will be 1.

82. What is 'std'?

    Default namespace defined by C++.

83. What is the full form of STL?

    Standard template library

84. What is 'cout'?

    cout is the object of ostream class. The stream 'cout' is by default connected to console output device.

85. What is 'cin'?

    cin is the object of istream class. The stream 'cin' is by default connected to console input device.

86. What is the use of the keyword 'using'?

    It is used to specify the namespace being used in.

87. If a pointer declared for a class, which operator can be used to access its class members?

    Arrow (->) operator can be used for the same

88. What is difference between including the header file with-in angular braces < > and double quotes " "

    If a header file is included with in < > then the compiler searches for the particular header file only with in the built in include path. If a header file is included with in " ", then the compiler searches for the particular header file first in the current working directory, if not found then in the built in include path

89. S++ or S=S+1, which can be recommended to increment the value by 1 and why?

    S++, as it is single machine instruction (INC) internally.

90. What is the difference between actual and formal parameters?

    The parameters sent to the function at calling end are called as actual parameters while at the receiving of the function definition called as formal parameters.

91. What is the difference between variable declaration and variable definition?

    Declaration associates type to the variable whereas definition gives the value to the variable.

92. Which key word is used to perform unconditional branching?

    goto.

93. Is 068 a valid octal number?

    No, it contains invalid octal digits.

94. What is the purpose of #undef preprocessor?

    It will be used to undefine an existing macro definition.

95. Can we nest multi line comments in a C++ code?

    No, we cannot.

96. What is a virtual destructor?

A virtual destructor ensures that the objects resources are released in the reverse order of the object being constructed w.r.t inherited object.

97. What is the order of objects destroyed in the memory?

The objects are destroyed in the reverse order of their creation.

98. What is a friend class?

A class members can gain accessibility over other class member by placing the class declaration prefixed with the keyword ‘friend’ in the destination class.

99. What is the difference between interpreters and compilers?

Interpreters read through source code and translate a program, turning the programmer's code, or program instructions, directly into actions. Compilers translate source code into an executable program that can be run at a later time.

100. How do you compile the source code with your compiler?

Every compiler is different. Be sure to check the documentation that came with your compiler.

101. What does the linker do?

The linker's job is to tie together your compiled code with the libraries supplied by your compiler vendor and other sources. The linker lets you build your program in pieces and then link together the pieces into one big program.

102. What are the steps in the development cycle?

Edit source code, compile, link, test, repeat.

103. What is the difference between the compiler and the preprocessor?

Each time you run your compiler, the preprocessor runs first. It reads through your source code and includes the files you've asked for, and performs other housekeeping chores. The preprocessor is discussed in detail on Day 18, "Object-Oriented Analysis and Design."

104. Why is the function main() special?

main() is called automatically, each time your program is executed.

105. What are the two types of comments, and how do they differ?

C++-style comments are two slashes (//), and they comment out any text until the end of

the line. C-style comments come in pairs (/* */), and everything between the matching pairs is commented out. You must be careful to ensure you have matched pairs.

106. Can comments be nested?

Yes, C++-style comments can be nested within C-style comments. You can, in fact, nest C-style comments within C++-style comments, as long as you remember that the C++-style comments end at the end of the line.

107. Can comments be longer than one line?

C-style comments can. If you want to extend C++-style comments to a second line, you must put another set of double slashes (//).

108. What are the differences between the function prototype and the function defi-nition?

The function prototype declares the function; the definition defines it. The prototype ends with a semicolon; the definition need not. The declaration can include the keyword inline and default values for the parameters; the definition cannot. The declaration need not include names for the parameters; the definition must.

109. Do the names of parameters have to agree in the prototype, definition, and call to the function?

No. All parameters are identified by position, not name.

110. If a function doesn't return a value, how do you declare the function?

Declare the function to return void.

111. If you don't declare a return value, what type of return value is assumed?

Any function that does not explicitly declare a return type returns int.

112. What is a local variable?

A local variable is a variable passed into or declared within a block, typically a function. It is visible only within the block.

113. What is scope?

Scope refers to the visibility and lifetime of local and global variables. Scope is usually established by a set of braces.

114. What is polymorphism?

Polymorphism is the ability to treat many objects of differing but related types without regard to their differences. In C++, polymorphism is accomplished by using class derivation and virtual functions.

115. Is the declaration of a class its interface or its implementation?

The declaration of a class is its interface; it tells clients of the class how to interact with the class. The implementation of the class is the set of member functions stored--usually in a related CPP file.

116. What is the difference between public and private data members?

Public data members can be accessed by clients of the class. Private data members can be accessed only by member functions of the class.

117. Can member functions be private?

Yes. Both member functions and member data can be private.

118. Can member data be public?

Although member data can be public, it is good programming practice to make it private and to provide public accessor functions to the data.

119. Do class declarations end with a semicolon? Do class method definitions?

Declarations end with a semicolon after the closing brace; function definitions do not.

120. What is the difference between the indirection operator and the address of oper-ator?

The indirection operator returns the value at the address stored in a pointer. The address of operator (&) returns the memory address of the variable.

121. What is the difference between a reference and a pointer?

A reference is an alias, and a pointer is a variable that holds an address. References cannot be null and cannot be assigned to.

122. When must you use a pointer rather than a reference?

When you may need to reassign what is pointed to, or when the pointer may be null.

123. What does new return if there is insufficient memory to make your new object?

A null pointer (0).

124. What is a constant reference?

This is a shorthand way of saying "a reference to a constant object."

125. What is the difference between passing by reference and passing a reference?

Passing by reference means not making a local copy. It can be accomplished by passing a reference or by passing a pointer.

126. When you overload member functions, in what ways must they differ?

Overloaded member functions are functions in a class that share a name but differ in the number or type of their parameters.

127. What is the difference between a declaration and a definition?

A definition sets aside memory, but a declaration does not. Almost all declarations are definitions; the major exceptions are class declarations, function prototypes, and typedef statements.

128. When is the copy constructor called?

Whenever a temporary copy of an object is created. This happens every time an object is passed by value.

129. When is the destructor called?

The destructor is called each time an object is destroyed, either because it goes out of scope or because you call delete on a pointer pointing to it.

130. How does the copy constructor differ from the assignment operator (=)?

The assignment operator acts on an existing object; the copy constructor creates a new one.

131. What is the this pointer?

The this pointer is a hidden parameter in every member function that points to the object itself.

132. How do you differentiate between overloading the prefix and postfix increments?

The prefix operator takes no parameters. The postfix operator takes a single int parameter, which is used as a signal to the compiler that this is the postfix variant.

133. Can you overload the operator+ for short integers?

No, you cannot overload any operator for built-in types.

134. Is it legal in C++ to overload operator++ so that it decrements a value in your class?

It is legal, but it is a bad idea. Operators should be overloaded in a way that is likely to be readily understood by anyone reading your code.

135. What return value must conversion operators have in their declaration?

None. Like constructors and destructors, they have no return values.

136. What is a v-table?

A v-table, or virtual function table, is a common way for compilers to manage virtual

functions in C++. The table keeps a list of the addresses of all the virtual functions and, depending on the runtime type of the object pointed to, invokes the right function.

137. What is a virtual destructor?

A destructor of any class can be declared to be virtual. When the pointer is deleted, the runtime type of the object will be assessed and the correct derived destructor invoked.

138. How do you show the declaration of a virtual constructor?

There are no virtual constructors.

139. How can you create a virtual copy constructor?

By creating a virtual method in your class, which itself calls the copy constructor.

140. How do you invoke a base member function from a derived class in which you've overridden that function?

Base::FunctionName();

141. How do you invoke a base member function from a derived class in which you have not overridden that function?

FunctionName();

142. If a base class declares a function to be virtual, and a derived class does not use the term virtual when overriding that class, is it still virtual when inherited by a third-generation class?

Yes, the virtuality is inherited and cannot be turned off.

143. What is the protected keyword used for?

protected members are accessible to the member functions of derived objects.

144. What is a down cast?

A down cast (also called "casting down") is a declaration that a pointer to a base class is to be treated as a pointer to a derived class.

145. What is the v-ptr?

The v-ptr, or virtual-function pointer, is an implementation detail of virtual functions. Each object in a class with virtual functions has a v-ptr, which points to the virtual function table for that class.

146. If a round rectangle has straight edges and rounded corners, your RoundRect class inherits both from Rectangle and from Circle, and they in turn both inherit from Shape, how many Shapes are created when you create a RoundRect?

If neither class inherits using the keyword virtual, two Shapes are created: one for Rectangle and one for Shape. If the keyword virtual is used for both classes, only one shared Shape is created.

147. If Horse and Bird inherit virtual public from Animal, do their constructors initialize the Animal constructor? If Pegasus inherits from both Horse and Bird, how does it initialize Animal's constructor?

Both Horse and Bird initialize their base class, Animal, in their constructors. Pegasus does as well, and when a Pegasus is created, the Horse and Bird initializations of Animal are ignored.

148. Declare a class Vehicle and make it an abstract data type.

```
class Vehicle
{
virtual void Move() = 0;
}
```

149. If a base class is an ADT, and it has three pure virtual functions, how many of these functions must be overridden in its derived classes?

None must be overridden unless you want to make the class non-abstract, in which case all three must be overridden.

150. Can static member variables be private?

Yes. They are member variables, and their access can be controlled like any other. If they are private, they can be accessed only by using member functions or, more commonly, static member functions.

151. Show the declaration for a static member variable.

```
static int itsStatic;
```

152. Show the declaration for a static function pointer.

```
static int SomeFunction();
```

153. Show the declaration for a pointer to function returning long and taking an integer parameter.

```
long (* function)(int);
```

154. How do you establish an is-a relationship?

With public inheritance.

155. How do you establish a has-a relationship?

With containment; that is, one class has a member that is an object of another type.

156. What is the difference between containment and delegation?

Containment describes the idea of one class having a data member that is an object of another type. Delegation expresses the idea that one class uses another class to accomplish a task or goal. Delegation is usually accomplished by containment.

157. What is the difference between delegation and implemented-in-terms-of?

Delegation expresses the idea that one class uses another class to accomplish a task or goal. Implemented-in-terms-of expresses the idea of inheriting implementation from another class.

158. What is a friend function?

A friend function is a function declared to have access to the protected and private members of your

class.

159. What is a friend class?

A friend class is a class declared so that all its member functions are friend functions of your class.

160. If Dog is a friend of Boy, is Boy a friend of Dog?

No, friendship is not commutative.

161. If Dog is a friend of Boy, and Terrier derives from Dog, is Terrier a friend of Boy?

No, friendship is not inherited.

162. If Dog is a friend of Boy and Boy is a friend of House, is Dog a friend of House?

No, friendship is not associative.

163. Where must the declaration of a friend function appear?

Anywhere within the class declaration. It makes no difference whether you put the declaration within the public:, protected:, or private: access areas.

164. What is the insertion operator and what does it do?

The insertion operator (<<) is a member operator of the ostream object and is used for writing to the output device. 70. What is the extraction operator and what does it do? The extraction operator (>>) is a member operator of the istream object and is used for writing to your program's variables.

165. What are the three forms of cin.get() and what are their differences?

The first form of get() is without parameters. This returns the value of the character found, and will return EOF (end of file) if the end of the file is reached.

The second form of cin.get() takes a character reference as its parameter; that character is filled with the next character in the input stream. The return value is an iostream object.

The third form of cin.get() takes an array, a maximum number of characters to get, and a terminating character. This form of get() fills the array with up to one fewer characters than the maximum (appending null) unless it reads the terminating character, in which case it immediately writes a null and leaves the terminating character in the buffer.

166. What is the difference between cin.read() and cin.getline()?

cin.read() is used for reading binary data structures.

getline() is used to read from the istream's buffer.

167. What is the default width for ouputting a long integer using the insertion operator?

Wide enough to display the entire number.

168. What is the return value of the insertion operator?

A reference to an istream object.

169. What parameter does the constructor to an ofstream object take?

The filename to be opened.

170. What does the ios::ate argument do?

ios::ate places you at the end of the file, but you can write data anywhere in the file.

171. What is an inclusion guard?

Inclusion guards are used to protect a header file from being included into a program more than once.

172. How do you instruct your compiler to print the contents of the intermediate file showing the effects of the preprocessor?

This quiz question must be answered by you, depending on the compiler you are using.

173. What is the difference between #define debug 0 and #undef debug?

#define debug 0 defines the term debug to equal 0 (zero). Everywhere the word debug is found, the character 0 will be substituted. #undef debug removes any definition of debug; when the word debug is found in the file, it will be left unchanged.

174. Why can't you call invariants() as the first line of your constructor?

The job of your constructor is to create the object. The class invariants cannot and should not exist before the object is fully created, so any meaningful use of invariants() will return false until the constructor is finished.

175. What is the difference between object-oriented programming and procedural programming?

Procedural programming focuses on functions separate from data. Object-oriented programming ties data and functionality together into objects, and focuses on the interaction among the objects.

176. To what does "event-driven" refer?

Event-driven programs are distinguished by the fact that action is taken only in response to some form of (usually external) simulation, such as a user's keyboard or mouse input.

177. What are the stages in the development cycle?

Typically, the development cycle includes analysis, design, coding, testing, programming, and interaction and feedback among these stages.

178. What is a rooted hierarchy?

A rooted hierarchy is one in which all the classes in the program derive directly or indirectly from a single base class.

179. What is a driver program?

A driver program is simply a function that is designed to exercise whatever objects and functions you are currently programming.

180. What is encapsulation?

Encapsulation refers to the (desirable) trait of bringing together in one class all the data and functionality of one discrete entity.

181. What is the difference between a template and a macro?

Templates are built into the C++ language and are type-safe. Macros are implemented by the preprocessor and are not type-safe.

182. What is the difference between the parameter to a template and the parameter to a function?

The parameter to the template creates an instance of the template for each type. If you create six template instances, six different classes or functions are created. The

parameters to the function change the behavior or data of the function, but only one function is created.

183. What is the difference between a type-specific template friend class and a general template friend class?

The general template friend function creates one function for every type of the parameterized class; the type-specific function creates a type-specific instance for each instance of the parameterized class.

184. Is it possible to provide special behavior for one instance of a template but not for other instances?

Yes, create a specialized function for the particular instance. In addition to creating Array::SomeFunction(), also create Array::SomeFunction() to change the behavior for integer arrays.

185. How many static variables are created if you put one static member into a template class definition?

One for each instance of the class.

186. What is an exception?

An exception is an object that is created as a result of invoking the keyword throw. It is used to signal an exceptional condition, and is passed up the call stack to the first catch statement that handles its type.

187. What is a try block?

A try block is a set of statements that might generate an exception.

188. What is a catch statement?

A catch statement has a signature of the type of exception it handles. It follows a try block and acts as the receiver of exceptions raised within the try block.

189. What information can an exception contain?

An exception is an object and can contain any information that can be defined within a user-created class.

190. When are exception objects created?

Exception objects are created when you invoke the keyword throw.

191. Should you pass exceptions by value or by reference?

In general, exceptions should be passed by reference. If you don't intend to modify the contents of the exception object, you should pass a const reference.

192. Will a catch statement catch a derived exception if it is looking for the base class?

Yes, if you pass the exception by reference.

193. If there are two catch statements, one for base and one for derived, which should come first?

catch statements are examined in the order they appear in the source code. The first catch statement whose signature matches the exception is used.

194. What does catch(...) mean?

catch(...) will catch any exception of any type.

195. What is a breakpoint?

A breakpoint is a place in the code where the debugger will stop execution.

196. What is the difference between strcpy() and strncpy()?

strcpy(char* destination, char* source) copies source to destination, and puts a null at the end of destination. destination must be large enough to accommodate source, or strcpy() will simply write past the end of the array. strncpy(char* destination char* source, int howmany) will write howmany bytes of source to destination, but will not put a terminating null.

197. What is the function to call to turn an ASCII string into a long?

atol()