1. Wrapping data and its related functionality into a single entity is known as \_\_\_\_\_\_\_\_\_\_\_\_\_  
a) Abstraction  
b) Encapsulation  
c) Polymorphism  
d) Modularity

Answer: b  
Explanation: In OOPs, the property of enclosing data and its related functions into a single entity(in C++ we call them classes) is called encapsulation.

2. How structures and classes in C++ differ?  
a) In Structures, members are public by default whereas, in Classes, they are private by default  
b) In Structures, members are private by default whereas, in Classes, they are public by default  
c) Structures by default hide every member whereas classes do not  
d) Structures cannot have private members whereas classes can have

Answer: a  
Explanation: Structure members are public by default whereas, class members are private by default. Both of them can have private and public members.

3. What does polymorphism in OOPs mean?  
a) Concept of allowing overriding of functions  
b) Concept of hiding data  
c) Concept of keeping things in different modules/files  
d) Concept of wrapping things into a single unit

Answer: a  
Explanation: In OOPs, Polymorphism is the concept of allowing a user to override functions either by changing the types or number of parameters passed.

4. Which concept allows you to reuse the written code?  
a) Encapsulation  
b) Abstraction  
c) Inheritance  
d) Polymorphism

Answer: c  
Explanation: Inheritance allows you to reuse your already written code by inheriting the properties of written code into other parts of the code, hence allowing you to reuse the already written code.

5. Which of the following explains Polymorphism?  
a)

int func(int, int);

float func1(float, float);

b)

int func(int);

int func(int);

c)

int func(float);

float func(int, int, char);

d)

int func();

int new\_func();

Answer: c  
Explanation: Polymorphism means overriding the same function by changing types or number of arguments. So we have only two options which has the same function names, but as one can observe that in one option types, name and number of parameters all are same which will lead to an error. Hence that is wrong so the option having same name and different types or number of parameters is correct.

6. Which of the following shows multiple inheritances?  
a) A->B->C  
b) A->B; A->C  
c) A,B->C  
d) B->A

Answer: c  
Explanation: In multiple inheritance, a single class is inherited from two classes. So in A,B->C, Class C is inherited from A and B, whereas in A->B->C, C from B and B from A called simple inheritance, in A->B; A->C, B and C are inherited from A which is called hierarchical inheritance.

7. C++ is \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) procedural programming language  
b) object oriented programming language  
c) functional programming language  
d) both procedural and object oriented programming language

Answer: d  
Explanation: C++ supports both procedural(step by step instruction) and object oriented programming(using concept of classes and objects).

8. What does modularity mean?  
a) Hiding part of program  
b) Subdividing program into small independent parts  
c) Overriding parts of program  
d) Wrapping things into single unit

Answer: b  
Explanation: Modularity means dividing a program into independent sub programs so that it can be invoked from other parts of the same program or any other program.

9. Which of the following feature of OOPs is not used in the following C++ code?

class A

{

int i;

public:

void print(){cout<<"hello"<<i;}

}

class B: public A

{

int j;

public:

void assign(int a){j = a;}

}

a) Abstraction  
b) Encapsulation  
c) Inheritance  
d) Polymorphism

Answer: d  
Explanation: As i and j members are private i.e. they are hidden from outer world therefore we have used the concept of abstraction. Next data members and there related functions are put together into single class therefore encapsulation is used. Also as class B is derived from A therefore Inheritance concept is used. But as no function is overloaded in any of the classes therefore, the concept of polymorphism is missing here.

10. Which of the following is correct?  
a) Base class pointer object cannot point to a derived class object  
b) Derived class pointer object cannot point to a base class object  
c) A derived class cannot have pointer objects  
d) A base class cannot have pointer objects

Answer: b  
Explanation: C++ does not allow a derived class pointer to point a base class pointer whereas Base class can point to a derived class object. Both base class and derived class can have pointer objects.

11. Out of the following, which is not a member of the class?  
a) Static function  
b) Friend function  
c) Constant function  
d) Virtual function

Answer: b  
Explanation: Friend function is not a member of the class. They are given the same access rights as the class member function have but they are not actual members of the class.

12. What is the other name used for functions inside a class?  
a) Member variables  
b) Member functions  
c) Class functions  
d) Class variables

Answer: b  
Explanation: Functions of a class are also known as member functions of a class.

13. Which of the following cannot be a friend?  
a) Function  
b) Class  
c) Object  
d) Operator function

Answer: c  
Explanation: Objects of any class cannot be made a friend of any other or same class whereas functions, classes and operator functions can be made a friend.

14. Why references are different from pointers?  
a) A reference cannot be made null  
b) A reference cannot be changed once initialized  
c) No extra operator is needed for dereferencing of a reference  
d) All of the mentioned

Answer: d  
Explanation: References cannot be made null whereas a pointer can be. References cannot be changed whereas pointers can be modified.  
Pointers need \* operator to dereference the value present inside it whereas reference does not need an operator for dereferencing.

15. Which of the following is an abstract data type?  
a) int  
b) float  
c) class  
d) string

Answer: c  
Explanation: Class is used as an abstract data type as it can be used to give implementation independent view whereas no other data type can be used to provide this.

16. Which of the following cannot be used with the virtual keyword?  
a) Class  
b) Member functions  
c) Constructors  
d) Destructors  
View Answer

Explanation: Virtual keyword cannot be used with constructors as constructors are defined to initialize an object of particular class hence no other class needs constructor of other class.

17. Which concept is used to implement late binding?  
a) Virtual functions  
b) Operator functions  
c) Constant functions  
d) Static functions

Answer: a  
Explanation: Virtual functions are used to implement the concept of late binding i.e. binding actual functions to their calls.

18. Which of the following is correct?  
a) C++ allows static type checking  
b) C++ allows dynamic type checking.  
c) C++ allows static member function to be of type const.  
d) C++ allows both static and dynamic type checking

Answer: d  
Explanation: C++ allows both static and dynamic type checking i.e. types are checked by the compiler.

19. Which members are inherited but are not accessible in any case?  
a) Private  
b) Public  
c) Protected  
d) Both private and protected

Answer: a  
Explanation: Private members of a class are inherited to the child class but are not accessible from the child class.

20. Which of the following is correct?  
a) Friend functions can access public members of a class  
b) Friend functions can access protected members of a class  
c) Friend functions can access private members of a class  
d) All of the mentioned

Answer: d  
Explanation: Friend functions can access any member of a class without caring about the type of member i.e. without caring whether it is private, protected or public.

21. Which of the following is used to make an abstract class?  
a) By using virtual keyword in front of a class declaration  
b) By using an abstract keyword in front of a class declaration  
c) By declaring a virtual function in a class  
d) By declaring a pure virtual function in a class

Answer: d  
Explanation: Abstract class should have at least one pure virtual function. Therefore to declare an abstract class one should declare a pure virtual function in a class.

22. What will be the output of the following C++ code?

#include <iostream>

using namespace std;

class A{

public:

A(){

cout<<"Constructor called**\n**";

}

~A(){

cout<<"Destructor called**\n**";

}

};

int main(int argc, char const \*argv[])

{

A \*a = new A[5];

delete a;

return 0;

}

a) “Constructor called” five times and then “Destructor called” five times  
b) “Constructor called” five times and then “Destructor called” once  
c) Error  
d) Segmentation fault

Answer: d  
Explanation: The program will result in segmentation fault as we are trying to delete only one pointer variable and leaving other variables as it is which will result in segmentation fault i.e. improper handling of memory.

23. What is the correct syntax of declaring array of pointers of integers of size 10 in C++?  
a) int arr = new int[10];  
b) int \*\*arr = new int\*[10];  
c) int \*arr = new int[10];  
d) int \*arr = new int\*[10];

Answer: b  
Explanation: As we have to declare an array of pointers of integers therefore we need double pointer array in which each element is collection pointers to integers. Therefore the correct syntax is int \*\*arr = new int\*[10];

24. What happens if the following C++ statement is compiled and executed?

int \*ptr = NULL;

delete ptr;

a) The program compiled successfully but throws an error during run-time  
b) The program gives a compile-time error  
c) The program is not semantically correct  
d) The program is compiled and executed successfully

Answer: d  
Explanation: The above statement is syntactically and semantically correct as C++ allows the programmer to delete a NULL pointer, therefore, the program is compiled and executed successfully.