



Unit 4 - Pointers:

MCQ OF UNIT-4

1. Which of the following class allows to declare only one object of it?

- a) Abstract class
- b) Virtual class
- c) Singleton class
- d) Friend class

[View Answer](#)

Answer: c

Explanation: Singleton class allows the programmer to declare only one object of it, If one tries to declare more than one object the program results into error.

2. Which of the following is not a type of Constructor?

- a) Friend constructor
- b) Copy constructor
- c) Default constructor
- d) Parameterized constructor

[View Answer](#)

Answer: a

Explanation: Friend function is not a constructor whereas others are a type of constructor used for object initialization.

3. 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

[View Answer](#)

Answer: b

Explanation: C++ does not allow a derived class pointer to point a



Unit 4 - Pointers:

base class pointer whereas Base class can point to a derived class object. Both base class and derived class can have pointer objects.

4. Out of the following, which is not a member of the class?

- a) Static function
- b) Friend function
- c) Constant function
- d) Virtual function

[View Answer](#)

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.

5. What is the other name used for functions inside a class?

- a) Member variables
- b) Member functions
- c) Class functions
- d) Class variables

[View Answer](#)

Answer: b

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

6. Which of the following cannot be a friend?

- a) Function
- b) Class
- c) Object
- d) Operator function

[View Answer](#)

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.

7. Why references are different from pointers?

- a) A reference cannot be made null



ALIGARH

Unit 4 - Pointers:

- b) A reference cannot be changed once initialized
- c) No extra operator is needed for dereferencing of a reference
- d) All of the mentioned

[View Answer](#)

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.

8. Which of the following provides a programmer with the facility of using object of a class inside other classes?

- a) Inheritance
- b) Composition
- c) Abstraction
- d) Encapsulation

[View Answer](#)

Answer: b

Explanation: The concept of using objects of one class into another class is known as Composition.

9. How many types of polymorphism are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: b

Explanation: There are two types of polymorphism in C++ namely run-time and compile-time polymorphisms.

10. How run-time polymorphisms are implemented in C++?

- a) Using Inheritance
- b) Using Virtual functions
- c) Using Templates



ALIGARH

Unit 4 - Pointers:

d) Using Inheritance and Virtual functions

[View Answer](#)

Answer: d

Explanation: Run-time polymorphism is implemented using Inheritance and virtual in which object decides which function to call.

11. How compile-time polymorphisms are implemented in C++?

a) Using Inheritance

b) Using Virtual functions

c) Using Templates

d) Using Inheritance and Virtual functions

[View Answer](#)

Answer: c

Explanation: Compile-time polymorphism is implemented using templates in which the types(which can be checked during compile-time) are used decides which function to be called.

12. Which of the following is an abstract data type?

a) int

b) float

c) class

d) string

[View Answer](#)

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.

13. Which concept means the addition of new components to a program as it runs?

a) Data hiding

b) Dynamic binding

c) Dynamic loading

d) Dynamic typing

[View Answer](#)



ALIGARH

Unit 4 - Pointers:

Answer: c

Explanation: Dynamic loading is the concept of adding new components to a program as it runs.

14. Which of the following explains the overloading of functions?

- a) Virtual polymorphism
- b) Transient polymorphism
- c) Ad-hoc polymorphism
- d) Pseudo polymorphism

[View Answer](#)

Answer: c

Explanation: Ad-hoc polymorphism is a type of polymorphism in which a function denotes heterogeneous implementation depending upon the types of argument.

15. Which of the following approach is used by C++?

- a) Top-down
- b) Bottom-up
- c) Left-right
- d) Right-left

[View Answer](#)

Answer: b

Explanation: C++ is an object-oriented language and OOL uses a bottom-up approach to solve/view a problem.

16. What does the following statement mean?

```
int (*fp) (char*);
```

- a) pointer to a pointer
- b) pointer to an array of chars
- c) pointer to function taking a char* argument and returns an int
- d) function taking a char* argument and returning a pointer to int

[View Answer](#)

Answer: C

Explanation: The (*fn) represents a pointer to a function and char* as



ALIGARH

Unit 4 - Pointers:

arguments and returning int from the function. So according to that, the above syntax represents a pointer to a function taking a char* as an argument and returning int.

17. The operator used for dereferencing or indirection is ____

- a) *
- b) &
- c) ->
- d) ->>

[View Answer](#)

Answer: A

Explanation: * is used as dereferencing operator, used to read value stored at the pointed address.

18. Choose the right option.

string* x, y;

- a) x is a pointer to a string, y is a string
- b) y is a pointer to a string, x is a string
- c) both x and y are pointers to string types
- d) y is a pointer to a string

[View Answer](#)

Answer: A

Explanation: * is to be grouped with the variables, not the data types.

19. Which one of the following is not a possible state for a pointer.

- a) hold the address of the specific object
- b) point one past the end of an object



ALIGARH

Unit 4 - Pointers:

- c) zero
- d) point to a type

[View Answer](#)

Answer: **D**

Explanation: A pointer can be in only 3 states a, b and c.

20. Which of the following is illegal?

- a) `int *ip;`
- b) `string s, *sp = 0;`
- c) `int i; double* dp = &i;`
- d) `int *pi = 0;`

[View Answer](#)

Answer: **C**

Explanation: `dp` is initialized with `int` value of `i`.

21. What will happen in the following C++ code snippet?

```
int a =100, b =200;
int *p = &a, *q = &b ;
p = q
```

- a) `b` is assigned to `a`
- b) `p` now points to `b`
- c) `a` is assigned to `b`
- d) `q` now points to `a`

[View Answer](#)

Answer: **B**

Explanation: Assigning to reference changes the object to which the reference is bound.



Unit 4 - Pointers:

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

```
#include <iostream>
using namespace std;
int main() {
    int a = 5, b = 10, c = 15;
    int *arr[] = {&a, &b, &c};
    cout << arr[1];
    return 0;
}
```

- a) 5
- b) 10
- c) 15
- d) it will return some random number

View Answer

Answer: **D**

Explanation: Array element cannot be address of auto variable. It can be address of static or extern variables.

23. The correct statement for a function that takes pointer to a float, a pointer to a pointer to a char and returns a pointer to a pointer to a integer is _____

- a) `int **fun(float**, char**)`
- b) `int *fun(float*, char*)`



ALIGARH

Unit 4 - Pointers:

- c) int **fun(float*, char**)
- d) int ***fun(*float, **char)

View Answer

Answer: C

Explanation: Function that takes pointer to a float, a pointer to a pointer to a char and returns a pointer to a pointer to an integer is int **fun(float*, char**).

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

```
#include <iostream>
using namespace std;
int main() {

    char arr[20];

    int i;

    for (i = 0; i < 10; i++)
        *(arr + i) = 65 + i;

    *(arr + i) = '\0';

    cout << arr;

    return 0;
}
```

- a) ABCDEFGHIJ
- b) AAAAAAAAAA
- c) JJJJJJJ
- d) AAAAAAJJJ



ALIGARH

Unit 4 - Pointers:

[View Answer](#)

Answer: **A**

Explanation: Each time we are assigning 65 + i. In first iteration i = 0 and 65 is assigned. So it will print from A to J.

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

```
#include <iostream>
using namespace std;
int main() {

    char *ptr;
    char str[] = 'abcdefg';
    ptr = str;
    ptr += 5;
    cout << ptr;

    return 0;
}
```

- a) fg
- b) cdef
- c) defg
- d) abcd

[View Answer](#)

Answer: **A**

Explanation: Pointer ptr points to string 'fg'. So it prints fg.



Unit 4 - Pointers:

26. What is the meaning of the following declaration?

```
int (*p [5]) ();
```

- a) p is pointer to function
- b) p is array of pointer to function
- c) p is pointer to such function which return type is the array
- d) p is pointer to array of function

View Answer

Answer: **B**

Explanation: In the above declaration the variable p is the array, not the pointer.

27. What is size of generic pointer in C++ (in 32-bit platform)?

- a) 2
- b) 4
- c) 8
- d) 0

View Answer

Answer: **B**

Explanation: Size of any type of pointer is 4 bytes in 32-bit platforms.

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

```
#include <iostream>

using namespace std;

int main() {

    int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
```



Unit 4 - Pointers:

```
cout << *(a[1] + 2) << * (*(a + 1) + 2) << [1[a]];

return 0;

}
```

- a) 15 18 21
- b) 21 21 21
- c) 24 24 24
- d) Compile time error

View Answer

Answer: **B**

Explanation: a[1][2] means $1 * (4) + 2 = 6$ th element of an array starting from zero.

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

```
#include <iostream>

using namespace std;

int main() {

    int i;

    const char *arr[] = {"c", "c++", "java", "VBA"};

    const char *(*ptr)[4] = &arr;

    cout << ++(*ptr)[2];

    return 0;

}
```



ALIGARH

Unit 4 - Pointers:

- a) ava
- b) java
- c) c++
- d) compile time error

[View Answer](#)

Answer: **A**

Explanation: In this program we are moving the pointer from first position to second position and printing the remaining value.

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

```
#include <iostream>
using namespace std;
int main() {
    int arr[] = {4, 5, 6, 7};
    int *p = (arr + 1);
    cout << *p;
    return 0;
}
```

- a) 4
- b) 5
- c) 6
- d) 7

[View Answer](#)



ALIGARH

Unit 4 - Pointers:

Answer: **B**

Explanation: In this program, we are making the pointer point to next value and printing it.

35. What will happen in the following C++ code snippet?

```
#include <iostream>
using namespace std;
int main() {

    int arr[] = {4, 5, 6, 7};
    int *p = (arr + 1);
    cout << arr;

    return 0;
}
```

- a) 4
- b) 5
- c) address of arr
- d) 7

[View Answer](#)

Answer: **C**

Explanation: As we counted to print only arr, it will print the address of the array.

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



Unit 4 - Pointers:

```
#include <iostream>
using namespace std;
int main() {

    int numbers[5];

    int *p;

    p = numbers; *p = 10;

    p++; *p = 20;

    p = numbers[2]; *p = 30;

    p = numbers + 3; *p = 40;

    p = numbers; *(p + 4) = 50;

    for (int n = 0; n < 5; n++){
        cout << numbers[n] << ", ";
    }

    return 0;
}
```

- a) 10,20,30,40,50,
- b) 1020304050
- c) compile error
- d) runtime error

[View Answer](#)

Answer: A

Explanation: In this program, we are just assigning a value to the array and printing it and immediately dereferencing it.



Unit 4 - Pointers:

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

```
#include <iostream>
using namespace std;
int main() {

    int arr[] = {4, 5, 6, 7};
    int *p = (arr + 1);
    cout << *arr + 9;

    return 0;
}
```

- a) 12
- b) 5
- c) 13
- d) error

[View Answer](#)

Answer: **C**

Explanation: In this program, we are adding the value 9 to the initial value of the array, So it's printing as 13.

39. The void pointer can point to which type of objects?

- a) int
- b) float
- c) double
- d) all of the mentioned

[View Answer](#)



Unit 4 - Pointers:

Answer: **D**

Explanation: Because it doesn't know the type of object it is pointing to, So it can point to all objects.

40. When does the void pointer can be dereferenced?

- a) when it doesn't point to any value
- b) when it cast to another type of object
- c) using delete keyword
- d) using shift keyword

[View Answer](#)

Answer: **B**

Explanation: By casting the pointer to another data type, it can be dereferenced from the void pointer.

41. What is Inheritance in C++?

- a) Wrapping of data into a single class
- b) Deriving new classes from existing classes
- c) Overloading of classes
- d) Classes with same names

[View Answer](#)

Answer: b

Explanation: Inheritance is the concept of OOPs in which new classes are derived from existing classes in order to reuse the properties of classes defined earlier.

42. How many specifiers are used to derive a class?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)



ALIGARH

Unit 4 - Pointers:

Answer: c

Explanation: There are 3 specifiers used to derive a class. They are private, protected and public.

43. Which specifier makes all the data members and functions of base class inaccessible by the derived class?

- a) private
- b) protected
- c) public
- d) both private and protected

[View Answer](#)

Answer: a

Explanation: Private access specifier is used to make all the data members and functions of the base class inaccessible.

advertisement

44. If a class is derived privately from a base class then

- a) no members of the base class is inherited
- b) all members are accessible by the derived class
- c) all the members are inherited by the class but are hidden and cannot be accessible
- d) no derivation of the class gives an error

[View Answer](#)

Answer: c

Explanation: Whenever a class is derived, all the members of the base class is inherited by the derived class but are not accessible by the derived class.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a, b;
```



Unit 4 - Pointers:

```
float d;  
public:  
    void change(int i){  
        a = i;  
    }  
    void value_of_a(){  
        cout<<a;  
    }  
};  
  
class B: private A  
{  
  
};  
  
int main(int argc, char const *argv[])  
{  
    B b;  
    cout<<sizeof(B);  
    return 0;  
}
```

- a) 8
- b) 12
- c) Error
- d) Segmentation fault

[View Answer](#)

Answer: b

Explanation: As class B is derived from class A and class A has three members with each of 4 bytes size hence size of B equal to $3 * 4 = 12$ bytes.

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

```
#include <iostream>  
#include <string>  
using namespace std;  
class A  
{  
    float d;
```



Unit 4 - Pointers:

```
public:
    int a;
    void change(int i){
        a = i;
    }
    void value_of_a(){
        cout<<a;
    }
};

class B: public A
{
    int a = 15;
public:
    void print(){
        cout<<a;
    }
};

int main(int argc, char const *argv[])
{
    B b;
    b.change(10);
    b.print();
    b.value_of_a();

    return 0;
}
```

- a) 1010
- b) 1510
- c) 1515
- d) 5110

[View Answer](#)

Answer: b

Explanation: When change() is called it sets parents class 'a' variable = 0. When print() is called then 'a' from class B is printed and wehn value_of_a() is called then 'a' from class A is printed.

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



Unit 4 - Pointers:

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    float d;
public:
    A(){
        cout<<"Constructor of class A\n";
    }
};

class B: public A
{
    int a = 15;
public:
    B(){
        cout<<"Constructor of class B\n";
    }
};

int main(int argc, char const *argv[])
{
    B b;
    return 0;
}
```

a)

Constructor of class A

Constructor of class B

b) Constructor of class A

c) Constructor of class B

d)

Constructor of class B

Constructor of class A



Unit 4 - Pointers:

[View Answer](#)

Answer: a

Explanation: When a derived class is declared it calls both its constructor and the base class constructor. It first calls the base class constructor and then its own constructor.

48. What is a virtual function in C++?

- a) Any member function of a class
- b) All functions that are derived from the base class
- c) All the members that are accessing base class data members
- d) All the functions which are declared in the base class and is re-defined/overridden by the derived class

[View Answer](#)

Answer: d

Explanation: Virtual function is a function that is declared inside the base class and is re-defined inside the derived class.

49. Which is the correct syntax of declaring a virtual function?

- a) virtual int func();
- b) virtual int func(){};
- c) inline virtual func();
- d) inline virtual func(){};

[View Answer](#)

Answer: a

Explanation: To make a function virtual function we just need to add virtual keyword at the starting of the function declaration.

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

```
#include <iostream>
#include <string>
using namespace std;
class A{
    float d;
public:
```



Unit 4 - Pointers:

```
virtual void func(){
    cout<<"Hello this is class A\n";
}
};

class B: public A{
    int a = 15;
public:
    void func(){
        cout<<"Hello this is class B\n";
    }
};

int main(int argc, char const *argv[])
{
    B b;
    b.func();
    return 0;
}
```

- a) Hello this is class B
- b) Hello this is class A
- c) Error
- d) Segmentation fault

[View Answer](#)

Answer: a

Explanation: Normal execution of the program and object calls func() from class B.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    float d;
public:
    virtual void func(){
        cout<<"Hello this is class A\n";
    }
}
```



Unit 4 - Pointers:

```
    }  
};  
  
class B: public A  
{  
    int a = 15;  
public:  
    void func(){  
        cout<<"Hello this is class B\n";  
    }  
};  
  
int main(int argc, char const *argv[])  
{  
    A *a;  
    a->func();  
    return 0;  
}
```

- a) Hello this is class A
- b) Hello this is class B
- c) Error
- d) Segmentation Fault

[View Answer](#)

Answer: d

Explanation: As object 'a' is a pointer object and we know every pointer needs to be initialised memory before use. Hence segmentation fault. Use `A *a = new A();` to initialise memory to the object.

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

```
#include <iostream>  
#include <string>  
using namespace std;  
class A  
{  
    float d;  
public:  
    virtual void func(){
```




Unit 4 - Pointers:

```
        cout<<"Hello this is class A\n";
    }
};

class B: public A
{
    int a = 15;
public:
    void func(){
        cout<<"Hello this is class B\n";
    }
};

int main(int argc, char const *argv[])
{
    A *a = new A();
    B b;
    a = &b;
    a->func();
    return 0;
}
```

- a) Hello this is class A
- b) Hello this is class B
- c) Error
- d) Segmentation Fault

[View Answer](#)

Answer: b

Explanation: As pointer object a is pointing to the object b hence the definition of virtual function defined inside the class B will be class. This is one of the use of virtual function.

53. Which statement is incorrect about virtual function.

- a) They are used to achieve runtime polymorphism
- b) They are used to hide objects
- c) Each virtual function declaration starts with the virtual keyword
- d) All of the mentioned

[View Answer](#)



ALIGARH

Unit 4 - Pointers:

Answer: b

Explanation: Virtual function are used to achieve runtime polymorphism by calling the right function during runtime. Their declaration starts with a virtual keyword.

54. The concept of deciding which function to invoke during runtime is called _____

- a) late binding
- b) dynamic linkage
- c) static binding
- d) both late binding and dynamic linkage

[View Answer](#)

Answer: d

Explanation: The concept of deciding which function to invoke during runtime is called late binding or dynamic linkage. Late binding because function binding to the object is done during runtime. Dynamic linkage because this binding is done during runtime.

55. What is a pure virtual function?

- a) A virtual function defined inside the base class
- b) A virtual function that has no definition relative to the base class
- c) A virtual function that is defined inside the derived class
- d) Any function that is made virtual

[View Answer](#)

Answer: b

Explanation: A virtual function that has no definition relative to the base class is called a pure virtual function.

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

```
#include <iostream>
#include <string>
using namespace std;
class Mammal
{
public:
    Mammal(){
        cout<<"I'm a Mammal\n";
    }
};
```



Unit 4 - Pointers:

```
    }
    ~Mammal(){}
};

class Human: public Mammal
{
public:
    Human(){
        cout<<"I'm a Human\n";
    }
    ~Human(){}
};

class Male: public Human
{
public:
    Male(){
        cout<<"I'm a Male\n";
    }
    ~Male(){}
};

class Female: public Human
{
public:
    Female(){
        cout<<"I'm a Female\n";
    }
    ~Female(){}
};

int main(int argc, char const *argv[])
{
    Male M;
    return 0;
}
```

a)

I'm a Mammal



ALIGARH

Unit 4 - Pointers:

I'm a Human

I'm a Male

advertisement

b)

I'm a Mammal

I'm a Human

I'm a Female

c)

I'm a Human

I'm a Male

d)

I'm a Mammal

I'm a Male

[View Answer](#)

Answer: a

Explanation: As the Male class is derived from Human class and Human class is derived from the Mammal class. Therefore when an object of Male is declared then three constructors will be called namely Mammal(), Human() and Male() in the given order.

58. What is the order of Constructors call when the object of derived class B is declared, provided class B is derived from class A?

a) Constructor of A followed by B



ALIGARH

Unit 4 - Pointers:

- b) Constructor of B followed by A
- c) Constructor of A only
- d) Constructor of B only

[View Answer](#)

Answer: a

Explanation: Firstly the Constructor of class A is called then class B because the Constructor of the base class is called before derived class.

60. What is the order of Destructors call when the object of derived class B is declared, provided class B is derived from class A?

- a) Destructor of A followed by B
- b) Destructor of B followed by A
- c) Destructor of A only
- d) Destructor of B only

[View Answer](#)

Answer: b

Explanation: Order of Destructor call is just reverse of the order of Destructors call. First, the destructor of the derived class is called then Destructor of the base class is called.

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

```
#include <iostream>
#include <string>
using namespace std;
class Mammal
{
public:
    virtual void Define(){
        cout<<"I'm a Mammal\n";
    }
};

class Human: public Mammal
{
public:
    void Define(){
```



Unit 4 - Pointers:

```
        cout<<"I'm a Human\n";
    }
};

class Male: public Human
{
public:
    void Define(){
        cout<<"I'm a Male\n";
    }
};

class Female: public Human
{
public:
    void Define(){
        cout<<"I'm a Female\n";
    }
};

int main(int argc, char const *argv[])
{
    Mammal *M;
    Male m;
    Female f;
    *M = m;
    M->Define();
    return 0;
}
```

- a) I'm a Male
- b) I'm a Mammal
- c) Error
- d) Segmentation Fault

[View Answer](#)

Answer: d

Explanation: As the Mammal pointer *M is not Initialized memory therefore program results into segmentation faults.

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



Unit 4 - Pointers:

```
#include <iostream>
#include <string>
using namespace std;

class Mammal
{
public:
    virtual void Define(){
        cout<<"I'm a Mammal\n";
    }
};

class Human: public Mammal
{
public:
    void Define(){
        cout<<"I'm a Human\n";
    }
};

class Male: public Human
{
public:
    void Define(){
        cout<<"I'm a Male\n";
    }
};

class Female: public Human
{
public:
    void Define(){
        cout<<"I'm a Female\n";
    }
};

int main(int argc, char const *argv[])
{
    Mammal *M = new Mammal();
    Male m;
    Female f;
```



ALIGARH

Unit 4 - Pointers:

```
*M = m;  
M->Define();  
M = &m;  
M->Define();  
return 0;  
}
```

a)

I'm a Male

I'm a Male

b)

I'm a Male

I'm a Mammal

c)

I'm a Mammal

I'm a Male

d)

I'm a Mammal

I'm a Mammal

[View Answer](#)

Answer: c

Explanation: There is a difference between pointer and references. Pointer stores the address of a variable so we need dereferencing operator to access the pointed variable whereas references are another name for that variable so we don't need any dereferencing operator, they are dereference by compiler itself therefore when we



ALIGARH

Unit 4 - Pointers:

are using pointer then Mammal class definition is called and when reference is used then Male class definition is used.

63. Virtual functions in C++ tells the compiler to perform _____ on such functions.

- a) static binding
- b) late binding
- c) compile time binding
- d) no binding

[View Answer](#)

Answer: b

Explanation: Virtual function in C++ adds the power of late binding by deciding the type of object during run-time.

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

```
#include <iostream>
#include <string>
using namespace std;
class Mammal
{
    public:
        virtual void Define(){
            cout<<"I'm a Mammal\n";
        }
};

class Human: public Mammal
{
    private:
        void Define(){
            cout<<"I'm a Human\n";
        }
};

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



Unit 4 - Pointers:

```
Mammal *M = new Mammal();
Human H;
M = &H;
M->Define();
return 0;
}
```

- a) Error
- b) Segmentation fault
- c) I'm a Human
- d) Garbage Value

[View Answer](#)

Answer: c

Explanation: Using base class pointer we can call private functions of derived by using virtual keyword because virtual function asks compiler performs late binding i.e. bind function at run-time and at run-time there is no checking of access specifiers. Hence it can access private members.

65. Which concept of OOPs is shown by Virtual Functions?

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

[View Answer](#)

Answer: c

Explanation: Virtual function allows us to give different definitions of the same function i.e. overloading of functions which is known as Polymorphism.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
public:
```



Unit 4 - Pointers:

```
virtual A(){
    cout<<"A's Constructor\n";
}
};

class B: public A
{
public:
    A(){
        cout<<"Present inside the class B\n";
    }
};

int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) A's Constructor
- b) Present inside the class B
- c) Error
- d) Segmentation fault

[View Answer](#)

Answer: c

Explanation: C++ does not allow programmers to make a constructor a virtual function.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
public:
    virtual static void show(){
        cout<<"class A\n";
    }
};
```



Unit 4 - Pointers:

```
class B: public A
{
    public:
        static void show(){
            cout<<"class B\n";
        }
};

int main(int argc, char const *argv[])
{
    A *a = new A();
    B b;
    a = &b;
    a->show();
    return 0;
}
```

- a) class B
- b) Error
- c) Segmentation fault
- d) class A

[View Answer](#)

Answer: b

Explanation: A function cannot be made virtual and static at the same time.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;
    public:
        A(){
            a = 0;
        }
        static void show(){
```



Unit 4 - Pointers:

```
        a++;
        cout<<a;
    }
};

class B: public A
{
    public:
};

int main(int argc, char const *argv[])
{
    B b;
    b.show();
    return 0;
}
```

- a) 1
- b) 0
- c) Segmentation fault
- d) Error

[View Answer](#)

Answer: d

Explanation: Non-static members of class cannot be used inside a static functions of class.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;
    public:
    A(){
        a = 0;
    }
    void show(){
```



Unit 4 - Pointers:

```
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B: public A
{
    public:
};

int main(int argc, char const *argv[])
{
    B b;
    b.show();
    return 0;
}
```

- a) a: 1
- b) a: 0
- c) Error
- d) Segmentation fault

[View Answer](#)

Answer: a

Explanation: Though the constructor of class A is not called to initialize variable 'a' but as we know whenever we create an object of the derived class, constructors of both base and derived classes are called hence variable 'a' is initialized and program runs perfectly.

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

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;
    public:
    A(){
        a = 0;
    }
}
```



Unit 4 - Pointers:

```
void show(){
    a++;
    cout<<"a: "<<a<<endl;
}
};

class B: private A
{
    public:
};

int main(int argc, char const *argv[])
{
    B b;
    b.show();
    return 0;
}
```

- a) Error
- b) Segmentation fault
- c) a: 1
- d) a: 0

[View Answer](#)

Answer: a

Explanation: As class B is derived privately from A hence all the members of class A cannot be accessible by the object of class B hence the program gives error.

71. What happens if the following C++ program is compiled?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;
    public:
    A(){
        a = 0;
    }
};
```



Unit 4 - Pointers:

```
    }
    void show(){
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B: private A
{
    public:
        void show(){
            show();
        }
};

int main(int argc, char const *argv[])
{
    B b;
    b.show();
    return 0;
}
```

- a) Error because of the conflicts between two show() function in class B
- b) Program will compile successfully
- c) Error due to self call in show() function
- d) Error because show() function from class A is derived privately

[View Answer](#)

Answer: b

Explanation: As the program is syntactically correct and as one show() function is in class A and other in class B therefore no conflicts in same name function. Therefore program compiles successfully.

72. Pick the correct statement.

- a) Virtual function can have different names in the base and derived class
- b) Virtual function cannot be applied in Multiple Inheritance classes
- c) Virtual function are different in definitions only



ALIGARH

Unit 4 - Pointers:

d) Virtual function does early binding

[View Answer](#)

Answer: c

Explanation: Virtual functions differ in definitions only, prototype are similar. They does the late binding. They are applicable to all types of inheritance.

73. How many types of access specifiers are provided in OOP (C++)?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: c

Explanation: Only 3 types of access specifiers are available. Namely, private, protected and public. All these three can be used according to the need of security of members.

74. Which among the following can be used together in a single class?

- a) Only private
- b) Private and Protected together
- c) Private and Public together
- d) All three together

[View Answer](#)

Answer: d

Explanation: All the classes can use any of the specifiers as needed. There is no restriction on how many of them can be used together.

75. Which among the following can restrict class members to get inherited?

- a) Private
- b) Protected
- c) Public
- d) All three

[View Answer](#)



ALIGARH

Unit 4 - Pointers:

Answer: a

Explanation: Private members of a class can't be inherited. These members can only be accessible from members of its own class only. It is used to secure the data.

advertisement

76. Which access specifier is used when no access specifier is used with a member of class (java)?

- a) Private
- b) Default
- c) Protected
- d) Public

[View Answer](#)

Answer: b

Explanation: Default access is used if the programmer doesn't specify the specifier. This acts in a similar way as that of private. But since nothing is specified we call it to default access.

77. Which specifier allows a programmer to make the private members which can be inherited?

- a) Private
- b) Default
- c) Protected
- d) Protected and default

[View Answer](#)

Answer: c

Explanation: Protected access is used to make the members private. But those members can be inherited. This gives both security and code reuse capability to a program.

78. Which among the following is false?

- a) Private members can be accessed using friend functions
- b) Member functions can be made private
- c) Default members can't be inherited
- d) Public members are accessible from other classes also

[View Answer](#)



ALIGARH

Unit 4 - Pointers:

Answer: c

Explanation: The default members can be inherited. Provided that they are in same package. It works in a little different way from private access specifier.

79. If a class has all the private members, which specifier will be used for its implicit constructor?

- a) Private
- b) Public
- c) Protected
- d) Default

[View Answer](#)

Answer: b

Explanation: The implicit constructor will always be public. Otherwise the class wouldn't be able to have instances. In turn, no objects will be created and the class can only be used for inheritance.

80. If class A has add() function with protected access, and few other members in public. Then class B inherits class A privately. Will the user will not be able to call _____ from the object of class B.

- a) Any function of class A
- b) The add() function of class A
- c) Any member of class A
- d) Private, protected and public members of class A

[View Answer](#)

Answer: d

Explanation: Class B object will not be able to call any of the private, protected and public members of class A. It is not only about the function add(), but all the members of class A will become private members of class B.

81. Which access specifier should be used in a class where the instances can't be created?

- a) Private default constructor
- b) All private constructors
- c) Only default constructor to be public



ALIGARH

Unit 4 - Pointers:

d) Only default constructor to be protected

[View Answer](#)

Answer: b

Explanation: All the constructors must be made private. This will restrict the instance of class to be made anywhere in the program. Since the constructors are private, no instance will be able to call them and hence won't be allocated with any memory space.

90. On which specifier's data, does the size of a class's object depend?

- a) All the data members are added
- b) Only private members are added
- c) Only public members are added
- d) Only default data members are added

[View Answer](#)

Answer: a

Explanation: All the data members are counted to calculate the size of an object of a class. The data member access specifier doesn't play any role here. Hence all the data size will be added.

91. If class B inherits class A privately. And class B has a friend function. Will the friend function be able to access the private member of class A?

- a) Yes, because friend function can access all the members
- b) Yes, because friend function is of class B
- c) No, because friend function can only access private members of friend class
- d) No, because friend function can access private member of class A also

[View Answer](#)

Answer: c

Explanation: The friend function of class B will not be able to access private members of class A. Since B is inheriting class A privately, the members will become private in class B. But private members of class A won't be inherited at all. Hence it won't be accessible.



Unit 4 - Pointers:

92. If an abstract class has all the private members, then _____

- a) No class will be able to implement members of abstract class
- b) Only single inheritance class can implement its members
- c) Only other enclosing classes will be able to implement those members
- d) No class will be able to access those members but can implement.

[View Answer](#)

Answer: a

Explanation: The classes which inherit the abstract class, won't be able to implement the members of abstract class. The private members will not be inherited. This will restrict the subclasses to implement those members.

93. Which access specifier should be used so that all the parent class members can be inherited and accessed from outside the class?

- a) Private
- b) Default or public
- c) Protected or private
- d) Public

[View Answer](#)

Answer: d

Explanation: All the members must be of public access. So that the members can be inherited easily. Also, the members will be available from outside the class.

94. Which access specifier is usually used for data members of a class?

- a) Private
- b) Default
- c) Protected
- d) Public

[View Answer](#)

Answer: a

Explanation: All the data members should be made private to ensure the highest security of data. In special cases we can use public or



ALIGARH

Unit 4 - Pointers:

protected access, but it is advised to keep the data members private always.

95. Which specifier should be used for member functions of a class?

- a) Private
- b) Default
- c) Protected
- d) Public

[View Answer](#)

Answer: d

Explanation: It is always advised that the member functions should be kept public so that those functions can be used from out of the class. This is usually done to ensure that the features provided by the class can be used at its maximum.