

**Set 1**

1. 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.

2. 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.

3. 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.

4. Answer: b

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

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

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

15. 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.

Campus Credential

## Set 2

1. Answer: a

Explanation: The member functions can be accessed inside the class only if they are private. The access is scope is limited to ensure the security of the private members and their usage.

2. Answer: b

Explanation: The private members are accessible within the class. There is no restriction on use of private members by public or protected members. All the members can access the private member functions of the class.

3. Answer: a

Explanation: The private member functions can never be accessed in the derived classes. The access specifiers is of maximum security that allows only the members of self class to access the private member functions.

4. Answer: c

Explanation: The function declaration must contain private keyword followed by the return type and function name. Private keyword is followed by normal function declaration.

5. Answer: a

Explanation: The private members doesn't have to have the keyword with each private member. We only have to specify the keyword private followed by single colon and then private member's are listed.

6. Answer: d

Explanation: The private keyword must be mentioned before each private member. Unlike the rule in C++ to specify private once and list all other private member's, in java all member declarations must be preceded by the keyword private.

7. Answer: d

Explanation: There are no conditions applied on the number of private member functions that can be declared in a class. Though the system may restrict use of too many functions depending on memory.

8. Answer: c

Explanation: Even the private member functions can be called outside the class. This is possible if address of the function is known. We can use the address to call the function outside the class.

9. Answer: a

Explanation: The nested class members can't be accessed in the enclosed class even though other members can be accessed. This is to ensure the class members security and not to go against the rules of private members.

10. Answer: c

Explanation: The protected members are available within the class. And are also available in derived classes. But these members are treated as private members for outside the class and inheritance structure. Hence can't be accessed.

11. Answer: a

Explanation: The private member functions can be accessed within the class. A public member function can be called which in turn calls the private member function. This maintains the security and adheres to the rules of private members.

12. Answer: b

Explanation: If private functions get accessed even by the parent class that will violate the rules of private members. If the functions can be accessed then the derived class security is hindered.

13. Answer: a

Explanation: The private members access from outside the class produce an error. The error states that the code at some line can't access the private members. And denies the access terminating the program.

14. Answer: d

Explanation: The reason given in c option is wrong. The proper reason that the main function should not be private is that it should be accessible in whole program. This makes the program flexible.

15. Answer: b

Explanation: The private members can access any standard input or output. There is no restriction

on access to any input or output stream. And since standard input can also be used hence only accessing the output stream is not true.

Campus Credential

## Set 3

1. Answer: c

Explanation: The most suitable definition would be that public member functions are accessible everywhere using object of the class. If derived classes are using those, derived class object can be used to call those functions.

2. Answer: d

Explanation: The public member functions can be called using object of the class. The members can't be declared outside the class as those would become non-member functions of the class. The functions have security as those can be accessed using the class object only.

3. Answer: c

Explanation: The public member functions gets into the same specifier in which the inheritance is done. If protected members are involved in public inheritance, still those remain protected in the derived class but public members become public on public inheritance and protected in protected inheritance.

4. Answer: b

Explanation: The public member functions declaration must be mentioned with the keyword public. The syntax given is used in java. Keyword public is followed by the usual function declaration.

5. Answer: a

Explanation: The syntax in C++ must contain the public keyword followed by a colon. Thereafter, all the public members can be declared. But in few other language, public have to be mentioned explicitly with each member.

6. Answer: b

Explanation: The public members in java must be preceded with the keyword public. It must be mentioned with each public member, unlike C++ where we mention it only once. In java, each member must have explicit declaration of specifier type.

7. Answer: d

Explanation: The number of public members that can

be defined in a class doesn't have any limit. Though the programmer should not use too many functions, instead should use another class for more specific functions to reduce the readability complexity.

8. Answer: c

Explanation: The public members can be accessed anywhere in the program using the object of the class. And if object pointer is used, then arrow operator is used to access class members. If normal object is used with arrow operator, an error will be generated.

9. Answer: a

Explanation: The closest definition is that any public member function of the nested class can be accessed with the help of enclosing class object. The nested class object pointer can be used only within the enclosing class. It's not mandatory to use the members of nested class only within the enclosing class.

10. Answer: b

Explanation: The condition given says that there is no return type hence we can call the function directly. The object name should be mentioned with a dot operator to access its class members. Then the function name with parameters, if required, can be given.

11. Answer: b

Explanation: If private inheritance is used then the class containing the function will be able to use the function with rules of whichever specifier is used. Then the derived class makes those function the private members of itself. This restricts the public members of parent class from further inheritance.

12. Answer: b

Explanation: The public members of the base class can be accessed from the derived class object only if public inheritance is used. If protected or private inheritance is used then those members become public/protected in derived class and hence won't be able to be called from object of derived class.

13. Answer: a

Explanation: If the function is called directly without

using any object then the compiler doesn't get to know that the function has to be called from a specific class. And if there are no global or in-scope function with same name then the compiler produces an error stating that the called function is undeclared.

14. Answer: a

Explanation: The `main()` function must always be public. This is because the whole function and the operating system that is out of the package have to access the main function throughout the program execution. Hence the `main()` function should be public so as to be available everywhere in the program.

15. Answer: d

Explanation: The public member function can access any private, protected and public member of its class. Not only public member function, any member function of a class can access each and every other member declared inside the class. Hence are flexible to program.

## Set 4

1. Answer: b

Explanation: The problems that might occur during execution of a program are known as exceptions. The exceptions are unexpected sometimes and can be predicted. Also, the exceptions should be always considered for a better program.

2. Answer: a

Explanation: The exceptions should be handled to prevent any abnormal termination of a program. The program should keep running even if it gets interrupted in between. The program should preferable show the error occurred and then retry the process or just continue the program further.

3. Answer: c

Explanation: The exceptions may arise because the input given by the user might not be of the same type that a program can manage. If the input is invalid the program gets terminated.

4. Answer: a

Explanation: The exceptions are produced when anything unexpected happened. The program might not be able to find a file in the target location and hence program produces an exceptions. The exception produced, then terminates the program.

5. Answer: d

Explanation: Any type of exception can be handled by using class Exceptions. An object of this class is created which can manipulate the exception data. The data can be used to display the error or to run the program further based on error produced.

6. Answer: c

Explanation: The exceptions class is having two other derived classes which are of runtime exception handler and for other type of exceptions handling. The runtime exception handler is used to handle the exceptions produced during run time and same with case of other exceptions.

7. Answer: a

Explanation: Two blocks that are used to check for

errors and to handle the errors are try and catch block. The code which might produce some exceptions is placed inside the try block and then the catch block is written to catch the error that is produced. The error message or any other processing can be done in catch block if the error is produced.

8. Answer: a

Explanation: The try block may or may not have any catch block. But a catch block can't be there in a program if there is no try block. It is like else-block can only be written if and only if if-block is present in the program.

9. Answer: d

Explanation: There is no limit on the number of catch blocks corresponding to a try block. This is because the error can be of any type and for each type, a new catch block can be defined. This is to

make sure all type of exceptions can be handled.

10. Answer: c

Explanation: Actually all the functions are available in throwable class. But the return type given in

the option is wrong. The function toString returns string value. Hence the return type must be a String and not a char.

11. Answer: a

Explanation: The object must be created of a specific class of which the error has occurred. If the type of error is unknown then we can use an object of class Exceptions. This object will be able to handle any kind of exception that a program might produce.

12. Answer: b

Explanation: The separate catch blocks for a single try block can be combined into a single catch block. All type of errors can be then handled in a single block. The type still have to be specified for the errors that might be produced.

13. Answer: d

Explanation: A pipe symbol can be used to separate different type of exceptions. The exceptions

should always be given in proper sequence to ensure that no code remains unreachable. If not done properly the code might never be used in a program.

14. Answer: c

Explanation: There is a specific class to handle each type of exceptions that might be produced in a

program. The input and output exceptions can be handled by an object of class IOExceptions. This

class handles all type of input and output exceptions.

15. Answer: d

Explanation: Sometimes there is a need to execute a set of code every time the program runs. Even

if the exception occurs and even if it doesn't, there can be some code that must be executed at end

of the program. That code is written in finally block. This block is always executed regardless of

exceptions occurring.



## Set 5

1. Answer: b

Explanation: Inheritance indicates the code reusability. Encapsulation and abstraction are meant to hide/group data into one element. Polymorphism is to indicate different tasks performed by a single entity.

2. Answer: c

Explanation: For the feature given above, the OOP feature used is Polymorphism. Example of polymorphism in real life is a kid, who can be a student, a son, a brother depending on where he is.

3. Answer: d

Explanation: It is Encapsulation, which groups different properties and functions of a real world entity into single element. Abstraction, on other hand, is hiding of functional or exact working of codes and showing only the things which are required by the user.

4. Answer: c

Explanation: Data must be declared using objects. Object usage is mandatory because it in turn calls its constructors, which in turn must have a class defined. If object is not used, it is violation of pure OOP concept.

5. Answer: a

Explanation: Platform independence is not feature of OOP. C++ supports OOP but it's not a platform independent language. Platform independence depends on programming language.

6. Answer: d

Explanation: Encapsulation is indicated by use of classes. Inheritance is shown by inheriting the student class into topper class. Polymorphism is not shown here because we have defined the constructor in topper class but that doesn't mean that default constructor is overloaded.

7. Answer: d

Explanation: All the features are violated because Inheritance and Encapsulation won't be implemented. Polymorphism and Abstraction is still possible in

some cases, but the main features like data binding, object use and etc won't be used hence use of class is must for OOP concept.

8. Answer: a

Explanation: There are 7 basic features that define whether a programming language is pure OOP or not. The 4 basic features are inheritance, polymorphism, encapsulation and abstraction. Further, one is, object use is must, secondly, message passing and lastly, Dynamic binding.

9. Answer: c

Explanation: The interaction between two object is called message passing feature. Data transfer is not feature of OOP. Also, message reading is not feature of OOP.

10. Answer: d

Explanation: Virtual Functions can be defined in any class using the keyword virtual. All the classes which inherit the class containing the virtual function, define the virtual function as required. Redefining the function on all the derived classes according to class and use represents polymorphism.

11. Answer: a

Explanation: The feature is operator overloading. There is not feature named operator overriding specifically. Function overloading and overriding doesn't give addition function to any operator.

12. Answer: b

Explanation: Function overriding doesn't illustrate polymorphism because the functions are actually different and their scopes are different. Function and operator overloading illustrate proper polymorphism. Virtual functions show polymorphism because all the classes which inherit virtual function, define the same function in different ways.

13. Answer: a

Explanation: Exception handling is feature of OOP as it includes classes concept in most of the cases. Also it may come handy while using inheritance.

14. Answer: d

Explanation: The language must follow all the rules of OOP to be called a purely OOP language. Even if a single OOP feature is not followed, then it's known to be a partially OOP language.

15. Answer: a

Explanation: It is always true as we have the facility of private and protected access specifiers. Also, only the public and global data is available globally or else program should have proper permission to access the private data.

Campus Credential

## Set 6

1. A) a

the defaults value of a static integer variable of a class in Java is 0.

2. A) b

The execution goes on like this:

```
int i = 0; // i becomes 0
```

```
i = 0 + i; // now, i becomes 1
```

```
i = 0 + 1; // perform addition and  
assign 1 to i.
```

3. A) a

Multiple inheritances mean one class inheriting from more super classes.

4. A) b

5. A) c

Final methods of the base class cannot be overridden in the derived class.

6. A) c

7. A) D

The fields in an interface are implicitly specified as both static and final.

8. A) c

Parameter values are passed by value in the calling of a method, and so a copy of the value is created in the method, and the original value is not affected by the method call.

9. A) c

An interface can extend another interface but not implement.

10. A) c

11. A) d

12. A) c

13. A) d

14. A) c

15. A) d

**Set 7**

16. A) e

17. A) d

18. A) d

19. A) e

20. A) a

21. A) a

22. A) c

23. A) b

24. A) b

25. A) a

26. A) d

27. A) c

28. A) b

29. A) d

30. A) b

**Set 8**

31. A) d

32. A) c

33. A) c

34. A) c

35. A) b

36. A) b

37. A) e

38. A) b

39. A) c

40. A) d

41. A) c

42. A) a

43. A) d

44. A) d

45. A) b

46. A) c

47. A) b

48. A) d

49. A) e

50. A) d

51. A) e

52. A) c

53. A) b

54. A) e

55. A) b

56. A) d

57. A) c

58. A) c

59. A) b

60. A) d

**Set 9**

### Set 10

1. Answer: A

In simple word, if a class contains an object of another class as a data member, then it is known as composition. For example, Class Y, have a class X's object as data member. Means, Y is composed of X.

```
class X {
    public:
        void f1() {

        }
};

class Y{
    X obj;//class object as a data member
    public:
        void f2() {

        }
};

Another example,we can take that a house is
composed of windows, door and bricks etc. So, class
House will look like below

class Door {

};

class Windows {

};

class Bricks {

};

class House{
    Door _d;
    Windows _w;
```

Bricks \_b;

public:

```
void showHouse() {

}

};
```

2. Answer: B

There is no concept of virtual constructor available in C++. However, virtual destructor is available in C++ language to maintain the destructor call from derived to base class. In polymorphic classes, if we don't use virtual destructor in base class then the derived class destructor will not be called that may cause resource leaks.

3. Answer: A

Inheritance represents an 'is a' relationship.

4. Answer: C

Compile time polymorphism feature is used i.e. function overloading in C++. Function overloading means, in a class, multiple functions with same name can be written with different signatures, return types etc.

5. Answer: C

Two types of Polymorphism are available in C++ object oriented programming i.e. compile time and run time polymorphism. Also, known as early binding and late binding respectively.

Compiler time polymorphism features in C++ language are function overloading, constructor and operator overloading etc. and run time polymorphism is function overriding in inheritance relationship.

6. Answer: B

In C++ object oriented programming, abstract class is used for the same, in which we have common or say generalized function in abstract base class and also may have pure virtual function in this class that forces derived classes to implement it.

7. Answer: A

Virtual destructor is used to maintain the hierarchy of destructor calls for polymorphic classes in inheritance. If we don't use it then it may cause resource leak or memory leak.

8. Answer: A

Inheritance feature is used for concept of code re-usability as in inheritance a class can inherit properties and functions of existing well written class.

Abstraction : Provide only necessary information to client code.

Encapsulation: Hide complexity. e.g. by wrapping private class data members by functions. By making internal function of a class private and using interfaces etc.