**********"Constructors"**********

1. Which among the following is called first, automatically, whenever an object is created?
a) Class
b) Constructor
c) New
d) Trigger
View Answer

Answer: b
Explanation: Constructors are the member functions which are called automatically whenever an object is created. It is a mandatory functions to be called for an object to be created as this helps in initializing the object to a legal initial value for the class.
2. Which among the following is not a necessary condition for constructors?
a) Its name must be same as that of class
b) It must not have any return type
c) It must contain a definition body
d) It can contains arguments
View Answer

Answer: c
Explanation: Constructors are predefined implicitly, even if the programmer doesn't define any of them. Even if the programmer declares a constructor, it's not necessary that it must contain some definition.
3. Which among the following is correct?
a) class student{ public: int student(){} };
b) class student{ public: void student (){} };
c) class student{ public: student{}{} };
d) class student{ public: student(){} };
View Answer

Answer: d
Explanation: The constructors must not have any return type. Also, the body may or may not contain any body. Defining default constructor is optional, if you are not using any other constructor.

4. In which access should a constructor be defined, so that object of the class can be created in any function?
a) Public
b) Protected
c) Private
d) Any access specifier will work
View Answer

Answer: a
Explanation: Constructor function should be available to all the parts of program where the object is to be created. Hence it is advised to define it in public access, so that any other function is able to create objects.
5. How many types of constructors are available for use in general (with respect to parameters)?
a) 2
b) 3
c) 4

d) 5
View Answer

Answer: a
Explanation: Two types of constructors are defined generally, namely, default constructor and parameterized constructor. Default constructor is not necessary to be defined always.
Check this: Information Science MCQs | Computer Science MCQs
6. If a programmer defines a class and defines a default value parameterized constructor inside it.
He has not defined any default constructor. And then he try to create the object without passing arguments, which among the following will be correct?
a) It will not create the object (as parameterized constructor is used)
b) It will create the object (as the default arguments are passed)
c) It will not create the object (as the default constructor is not defined)
d) It will create the object (as at least some constructor is defined)
View Answer

Answer: b
Explanation: It will create the object without any problem, because the default arguments use the default value if no value is passed. Hence it is equal to default constructor with zero parameters. But it will not create the object if signature doesn't match.
7. Default constructor must be defined, if parameterized constructor is defined and the object is to be created without arguments.
a) True
b) False
View Answer

Answer: a
Explanation: If the object is create without arguments and only parameterized constructors are used, compiler will give an error as there is no default constructor defined. And some constructor must be called so as to create an object in memory.
8. If class C inherits class B. And B has inherited class A. Then while creating the object of class C, what will be the sequence of constructors getting called?
a) Constructor of C then B, finally of A
b) Constructor of A then C, finally of B
c) Constructor of C then A, finally B
d) Constructor of A then B, finally C
View Answer

Answer: d
Explanation: While creating the object of class C, its constructor would be called by default. But, if the class is inheriting some other class, firstly the parent class constructor will be called so that all the data is initialized that is being inherited.
9. In multiple inheritance, if class C inherits two classes A and B as follows, which class constructor will be called first?

```
class A{ };
class B{ };
class C: public A, public B{  };
```

a) A()
b) B()
c) C()
d) Can't be determined

View Answer

Answer: a
Explanation: Constructor of class A will be called first. This is because the constructors in multiple inheritance are called in the sequence in which they are written to be inherited. Here A is written first, hence it is called first.
10. Which among the following is true for copy constructor?
a) The argument object is passed by reference
b) It can be defined with zero arguments
c) Used when an object is passed by value to a function
d) Used when a function returns an object
View Answer

Answer: b
Explanation: It can't be defined with zero number of arguments. This is because to copy one object to another, the object must be mentioned so that compiler can take values from that object.
11. If the object is passed by value to a copy constructor?
a) Only public members will be accessible to be copied
b) That will work normally
c) Compiler will give out of memory error
d) Data stored in data members won't be accessible
View Answer

Answer: c
Explanation: Compiler runs out of memory. This is because while passing the argument by value, a constructor of the object will be called. That in turn called another object constructor for values, and this goes on. This is like a constructor call to itself, and this goes on infinite times, hence it must be passed by reference, so that the constructor is not called.
12. Which object will be created first?

```
class student
{
    int marks;
};
student s1, s2, s3;
```

a) s1 then s2 then s3
b) s3 then s2 then s1
c) s2 then s3 then s1
d) all are created at same time
View Answer

Answer: a
Explanation: The objects are created in the sequence of how they are written. This happens because the constructors are called in the sequence of how the objects are mentioned. This is done in sequence.
13. Which among the following helps to create a temporary instance?
a) Implicit call to a default constructor
b) Explicit call to a copy constructor
c) Implicit call to a parameterized constructor
d) Explicit call to a constructor
View Answer

Answer: d
Explanation: Explicit call to a constructor can let you create a temporary instance. This is because the temporary instances doesn't have

any name. Those are deleted from memory as soon as their reference is removed.

14. Which among the following is correct for the class defined below?

```
class student
{
    int marks;
    public: student(){}
    student(int x)
    {
         marks=x;
    }
};
main()
{
    student s1(100);
    student s2();
    student s3=100;
    return 0;
}
```

a) Object s3, syntax error
b) Only object s1 and s2 will be created
c) Program runs and all objects are created
d) Program will give compile time error
View Answer

Answer: c
Explanation: It is a special case of constructor with only 1 argument. While calling a constructor with one argument, you are actually implicitly creating a conversion from the argument type to the type of class. Hence you can directly specify the value of that one argument with assignment operator.

15. For constructor overloading, each constructor must differ in _____ and _____
a) Number of arguments and type of arguments
b) Number of arguments and return type
c) Return type and type of arguments
d) Return type and definition
View Answer

Answer: a
Explanation: Each constructor must differ in the number of arguments it accepts and the type of arguments. This actually defines the constructor signature. This helps to remove the ambiguity and define a unique constructor as required.

**********Types of Constructors"**********

1. How many types of constructors are available, in general, in any language?
a) 2
b) 3
c) 4
d) 5
View Answer

Answer: b

Explanation: There are 3 types of constructors in general, namely, default constructors, parameterized constructors and copy constructors. Default one is called whenever an object is created without arguments.
2. Choose the correct option for the following code.

```
class student
{
    int marks;
}
student s1;
student s2=2;
```

a) Object s1 should be passed with argument
b) Object s2 should not be declared
c) Object s2 will not be created, but program runs
d) Program gives compile time error
View Answer

Answer: d
Explanation: The object s2 can be assigned with one value only if a single argument constructor is defined in class, but here, it can't be done as no constructor is defined. Hence every object must be declare or created without using arguments.

3. Which constructor is called while assigning some object with another?
a) Default
b) Parameterized
c) Copy
d) Direct assignment is used
View Answer

Answer: c
Explanation: Copy constructor is used while an object is assigned with another. This is mandatory since we can't decide which member should be assigned to which member value. By using copy constructor, we can assign the values in required form.

4. It's necessary to pass object by reference in copy constructor because _____
a) Constructor is not called in pass by reference
b) Constructor is called in pass by reference only
c) It passes the address of new constructor to be created
d) It passes the address of new object to be created
View Answer

Answer: a
Explanation: Object must be passed by reference to copy constructor because constructor is not called in pass by reference. Otherwise, in pass by value, a temporary object will be created which in turn will try to call its constructor that is already being used. This results in creating infinite number of objects and hence memory shortage error will be shown.
5. Which specifier applies only to the constructors?
a) Public
b) Protected
c) Implicit
d) Explicit
View Answer

Answer: d
Explanation: The keyword explicit can be used while defining the constructor only. This is used to suppress the implicit call to the constructor. It ensures that the constructors are being called with the default syntax only (i.e. only by using object and constructor name).
6. Which among the following is true?
a) Default constructor can't be defined by the programmer
b) Default parameters constructor isn't equivalent to the default constructor
c) Default constructor can be called explicitly
d) Default constructor is and always called implicitly only
View Answer

Answer: c
Explanation: Default constructors can be called explicitly anytime. They are specifically used to allocate memory space for the object in memory, in general. It is not necessary that these should always be called implicitly.
7. Which type of constructor can't have a return type?
a) Default
b) Parameterized
c) Copy
d) Constructors don't have a return type
View Answer

Answer: d
Explanation: Constructors don't return any value. Those are special functions, whose return type is not defined, not even void. This is so because the constructors are meant to initialize the members of class and not to perform some task which can return some value to newly created object.
8. Why do we use static constructors?
a) To initialize the static members of class
b) To initialize all the members with static value
c) To delete the static members when not required
d) To clear all the static members initialized values
View Answer

Answer: a
Explanation: Static constructors help in initializing the static members of the class. This is provided because the static members are not considered to be property of the object, rather they are considered as the property of class.
9. When and how many times a static constructor is called?
a) Created at time of object destruction
b) Called at first time when an object is created and only one time
c) Called at first time when an object is created and called with every new object creation
d) Called whenever an object go out of scope
View Answer

Answer: b
Explanation: Those are called at very first call of object creation. That is called only one time because the value of static members must be retained and continued from the time it gets created.
10. Which among the following is true for static constructor?
a) Static constructors are called with every new object
b) Static constructors are used initialize data members to zero always
c) Static constructors can't be parameterized constructors

d) Static constructors can be used to initialize the non-static members also
View Answer

Answer: c
Explanation: Static constructors can't be parameterized constructors. Those are used to initialize the value of static members only. And that must be a definite value. Accepting arguments may make it possible that static members loses their value with every new object being created.
11. Within a class, only one static constructor can be created.
a) True
b) False
View Answer

Answer: a
Explanation: Since the static constructors are can't be parameterized, they can't be overloaded. Having this case, only one constructor will be possible to be created in a local scope, because the signature will always be same and hence it will not be possible to overload static constructor.
12. Default constructor initializes all data members as _____
a) All numeric member with some garbage values and string to random string
b) All numeric member with some garbage values and string to null
c) All numeric member with zero and strings to random value
d) All numeric member with zero and strings to null
View Answer

Answer: d
Explanation: Default constructor, which even the programmer doesn't define, always initialize the values as zero if numeric and null if string. This is done so as to avoid the accidental values to change the conditional statements being used and similar conditions.
13. When is the static constructor called?
a) After the first instance is created
b) Before default constructor call of first instance
c) Before first instance is created
d) At time of creation of first instance
View Answer

Answer: c
Explanation: The static constructor is called before creation of the first instance of that class. This is done so that even the first instance can use the static value of the static members of the class and manipulate as required.
14. If constructors of a class are defined in private access, then _____
a) The class can't be inherited
b) The class can be inherited
c) Instance can be created only in another class
d) Instance can be created anywhere in the program
View Answer

Answer: a
Explanation: If the constructors are defined in private access, then the class can't be inherited by other classes. This is useful when the class contains static members only. The instances can never be created.
15. Which among the following is correct, based on the given code below?

class student

```
{
     int marks;
     public : student()
     {
          cout<<"New student details can be added now";
     }
};
student s1;
```

a) Cout can't be used inside the constructor
b) Constructor must contain only initializations
c) This program works fine
d) This program produces errors
View Answer

Answer: c
Explanation: This program will work fine. This is because it is not mandatory that a constructor must contain only initialization only. If you want to perform a task on each instance being created, that code can be written inside the constructor

**********"Execution of Constructor or Destructor"**********

1. Which among the following best describes the constructors?
a) A function which is called whenever an object is referenced
b) A function which is called whenever an object is created to initialize the members
c) A function which is called whenever an object is assigned to copy the values
d) A function which is called whenever an object is to be given values for members
View Answer

Answer: b
Explanation: The constructors are special type of functions which are called whenever an object is created. This is to initialize the data members of the class. The constructor allocates memory space for all the data members.
2. Which among the following best describes destructor?
a) A function which is called just before the objects are destroyed
b) A function which is called after each reference to the object
c) A function which is called after termination of the program
d) A function which is called before calling any member function
View Answer

Answer: a
Explanation: The Destructors are special functions which are called just before an object is destroyed. This functions is responsible to free all the allocated resources to the object. Objects are destroyed whenever those go out of scope.
3. Which among the following represents correct constructor?
a) ()classname
b) ~classname()
c) -classname()
d) classname()
View Answer

Answer: d
Explanation: The constructors must contain only the class name. The class name is followed by the blank parenthesis or we can have parameters if some values are to be passed.
Note: Join free Sanfoundry classes at Telegram or Youtube
advertisement

4. Which among the following is correct syntax for the destructors?
a) classname()
b) ()classname
c) ~classname()
d) -classname()
View Answer

Answer: c
Explanation: The destructor must have same name as that of the corresponding class. The class name should be preceded by the tilde symbol (~).
5. Which among the following is true?
a) First the constructor of parent classes are called in sequence of inheritance
b) First the constructor of child classes are called in the sequence of inheritance
c) First constructor called is of the object being created
d) Constructors are called randomly
View Answer

Answer: a
Explanation: First the constructor of parent class are called. The order in which the parent class constructors are called is same in the sequence of inheritance used.
Take Object Oriented Programming Mock Tests - Chapterwise!
Start the Test Now: Chapter 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
6. What is the sequence of destructors call?
a) Same order as that of the constructors call
b) Random order
c) According to the priority
d) Revere of the order of constructor call
View Answer

Answer: d
Explanation: The destructors are called in the reverse order as that of the constructors being called. This is done to ensure that all the resources are released in sequence. That is, the derived class destructors will be called first.
7. The destructors _____
a) Can have maximum one argument
b) Can't have any argument
c) Can have more than one argument
d) Can't have more than 3 arguments
View Answer

Answer: b
Explanation: The destructors doesn't have any arguments. The destructors have to be called implicitly whenever an object goes out of scope. The user can't pass argument to the implicit call.
8. Destructor calls _____ (C++)
a) Are only implicit
b) Are only explicit
c) Can be implicit or explicit

d) Are made at end of program only
View Answer

Answer: c
Explanation: The destructors are usually called implicitly whenever an object goes out of scope. The destructors can also be called explicitly if required. The call must be made, implicitly or explicitly.
9. Number of destructors called are _____
a) Always equal to number of constructors called
b) Always less than the number of constructors called
c) Always greater than the number of constructors called
d) Always less than or equal to number of constructors
View Answer

Answer: a
Explanation: Destructor will be called only to free the resources allocated for an object. The resources are allocated only the constructor for an object is called.
10. For explicit call _____
a) The destructor must be private
b) The destructor must be public
c) The destructor must be protected
d) The destructor must be defined outside the class
View Answer

Answer: b
Explanation: The destructor must be public for explicit calls. If it is made private or protected then it won't be accessible outside the class. There is no restriction of definition the destructor outside the class.
11. If a class have 4 constructors then it must have 4 destructors also.
a) True
b) False
View Answer

Answer: b
Explanation: Even if the class have 4 constructors, only one would be used. And only one destructor is allowed.
12. Which among the following is true for destructors?
a) Destructors can be overloaded
b) Destructors can be define more than one time
c) Destructors can't be overloaded
d) Destructors are overloaded in derived classes
View Answer

Answer: c
Explanation: The destructors can never be overloaded. The destructors doesn't have arguments. And to get overloaded, they must have different signature. This is not possible if arguments are not allowed.
13. The constructor _____
a) Have a return type
b) May have a return type
c) Of derived classes have return type
d) Doesn't have a return type
View Answer

Answer: d
Explanation: The constructors doesn't have any return type. The constructors are intended to allocate the resources for the object. Memory spaces are to be finalized.
14. The destructors _____

a) Have a return type
b) May have a return type
c) Of derived classes have return type
d) Doesn’t have a return type
View Answer

Answer: d
Explanation: The destructors are intended to free the memory space. And all the resources that were allocated for the object. The return value is not supported since only memory has to be made free.
15. The destructor can be called before the constructor if required.
a) True
b) False
View Answer

Answer: b
Explanation: The destructors can be called only after the constructor calls. It is not a mandatory rule but the deletion can only take place if there is something created using the constructor.

********** ” Destructors”**********

1. Which among the following describes a destructor?
a) A special function that is called to free the resources, acquired by the object
b) A special function that is called to delete the class
c) A special function that is called anytime to delete an object
d) A special function that is called to delete all the objects of a class
View Answer

Answer: a
Explanation: It is used to free the resources that the object might had used in its lifespan. The destructors are called implicitly whenever an object’s life ends.
2. When a destructor is called?
a) After the end of object life
b) Anytime in between object’s lifespan
c) At end of whole program
d) Just before the end of object life
View Answer

Answer: d
Explanation: The destructor is called just before the object go out of scope or just before its life ends. This is done to ensure that all the resources reserved for the object are used and at last, are made free for others.
3. Which among the following is correct for abstract class destructors?
a) It doesn’t have destructors
b) It has destructors
c) It may or may not have destructors
d) It contains an implicit destructor
View Answer

Answer: a
Explanation: It doesn’t have destructors. Since an abstract class don’t have constructors, and hence can’t have instances. Having this case, the

abstract classes don’t have destructors too, because that would be of no use here.
Note: Join free Sanfoundry classes at Telegram or Youtube
advertisement

4. If in multiple inheritance, class C inherits class B, and Class B inherits class A. In which sequence are their destructors called if an object of class C was declared?
a) ~C() then ~B() then ~A()
b) ~B() then ~C() then ~A()
c) ~A() then ~B() then ~C()
d) ~C() then ~A() then ~B()
View Answer

Answer: a
Explanation: The destructors are always called in the reverse order of how the constructors were called. Here class A constructor would have been created first if Class C object is declared. Hence class A destructor is called at last.
5. Choose the correct sequence of destructors being called for the following code.


```
class A{   };
class B{   };
class C: public A, public B{   };
```
a) ~A(), ~B(), ~C()
b) ~B(), ~C(), ~A()
c) ~A(), ~C(), ~B()
d) ~C(), ~B(), ~A()
View Answer

Answer: d
Explanation: In multiple inheritance, the constructors are called in the sequence of how they are written in inheritance sequence. And the destructors will be called in the reverse order. This can be cross verified just by printing a message from each destructor defined in classes.
6. When is the destructor of a global object called?
a) Just before end of program
b) Just after end of program
c) With the end of program
d) Anytime when object is not needed
View Answer

Answer: a
Explanation: This is because the lifespan of global object is from start of the program, till the end of the program. And hence program end is the end of global object too. Just before the end of program, the destructor will be called to free the acquired resources by the objects.
7. How the constructors and destructors can be differentiated?
a) Destructor have a return type but constructor doesn’t
b) Destructors can’t be defined by the programmer, but constructors can be defined
c) Destructors are preceded with a tilde (~) symbol, and constructor doesn’t
d) Destructors are same as constructors in syntax
View Answer

Answer: c
Explanation: The destructors are preceded with the tilde (~) symbol. The name is same as that of the class. These also doesn't have any return type.
8. Destructors doesn't accept parameters.
a) True
b) False
View Answer

Answer: a
Explanation: The destructors doesn't accept the arguments. Those are just used to free up the resources.
9. Destructors can be _____
a) Abstract type
b) Virtual
c) Void
d) Any type depending on situation
View Answer

Answer: b
Explanation: The destructors can be virtual. It is actually advised to keep the destructors virtual always. This is done to suppress the problems that may arise if inheritance is involved.
10. Global destructors execute in _____ order after main function is terminated.
a) Sequential
b) Random
c) Reverse
d) Depending on priority
View Answer

Answer: c
Explanation: The destructors are always called in reverse order no matter which destructor it is. This is done to ensure that all the resources are able to get free. And no resource is kept busy.
11. When is it advised to have user defined destructor?
a) When class contains some pointer to memory allocated in class
b) When a class contains static variables
c) When a class contains static functions
d) When a class is inheriting another class only
View Answer

Answer: a
Explanation: This is always advised to have user defined destructor when pointers are involved in class. This is usually done to ensure that the memory, that was allocated dynamically, gets free after use and doesn't cause memory leak.
12. Which among the following is correct for the destructors concept?
a) Destructors can be overloaded
b) Destructors can have only one parameter at maximum
c) Destructors are always called after object goes out of scope
d) There can be only one destructor in a class
View Answer

Answer: d
Explanation: This is so because the destructors can't be overloaded. And the destructor must have the same name as that of class with a tilde symbol preceding the name of the destructor. Hence there can be only one destructor in a class. Since more than one function with same name and signature can't be present in same scope.

13. Which class destructor will be called first, when following code go out of scope?

```
class A{  };
class B{  };
class C: public B{  };
A a;
B b;
C c;
```

a) ~A()
b) ~B()
c) ~C()
d) ~B() and ~C()
View Answer

Answer: c
Explanation: The constructor that would have created at last, its destructor will be called first when the code goes out of scope. This will help the program to manage the resources more efficiently.

14. When an object is passed to a function, its copy is made in the function and then _____

a) The destructor of the copy is called when function is returned
b) The destructor is never called in this case
c) The destructor is called but it is always implicit
d) The destructor must be user defined
View Answer

Answer: a
Explanation: When an object is passed to a function, its copy is made in the function. This copy acts as a real object till the function is live. When the function is returned, the copy's destructor is called to free the resources held by it.

15. What happens when an object is passed by reference?

a) Destructor is not called
b) Destructor is called at end of function
c) Destructor is called when function is out of scope
d) Destructor is called when called explicitly
View Answer

Answer: a
Explanation: The destructor is never called in this situation. The concept is that when an object is passed by reference to the function, the constructor is not called, but only the main object will be used. Hence no destructor will be called at end of function.