Constructors

1. **Can we have two constructors with the same name, same number of arguments and same type of arguments?**
   Yes. Provided the order is different.

2 **What happens if we don't explicitly provide a constructor in our class?**
   Java compiler adds default constructor(zero parameterized constructor) inside which a call to super class constructor is present.

3 **Who inserts a no argument constructor into our class?**
   Java compiler.

4 **When does the no argument constructor get inserted into our class?**
   If the class does not contain any other constructor.

5 **What is the no argument constructor also called as?**
   Default constructor or zero parameterized constructor.

6 **Whom does the no argument constructor of our class implicitly call?**
   Super class constructor(Parent class constructor).

7 **Can we use access modifier in a constructor declaration?**
   Yes. Any access modifiers such as private , public , default and protected can be associated with a constructor.

8 **What is the difference between parameters and arguments?**
   Dog d = new Dog("Lobo", "Bulldog", 1234);
   Lobo, Bulldog, 12324 are called as arguments (Actual Parameters).

   public Dog(String name, String breed, int cost)
   name, breed, cost are called as parameters (Formal Parameters).

9 What relation must hold good between parameters and arguments?

The number of arguments and the number of parameters should be same. Datatypes of arguments and parameters should be the same and also order should match.

10 Can we pass a method into another method as a parameter?

Yes.

11 How can we pass a method into another method as a parameter?

Using lambda expression.

12 When does a parameter shadow the class field?

When parameter names and instance variable's names are same then it would shadow the class field.

13 How can shadowing problem be resolved?

Using "this" keyword.

14Is it compulsory to have a constructor within a class?

Yes. It is compulsory for a class to have a constructor. If the programmer does not manually provide a constructor, then java compiler automatically adds a default constructor(zero parameterized constructor).

15Can we overload constructors?

Yes.

16What is meant by constructor chaining?

The process of child class constructor calling its super class(parent class) constructor using super() method is called as constructor chaining.

17 What is the return type of a constructor?

Constructors do not have return type.

18 What does the constructor return?

It returns the address of an object which would normally collected in a reference variable.

19Does the compiler include default constructor if the class already contains a user defined constructor?

No.

20 How can local chaining of constructors be achieved?

Using this() method.

21 Which construct must be used to achieve local chaining? What is its limitation?

Using this() method local chaining can be achieved.
The limitation of this() method is that, in order to use this() method minimum two constructors must be present within a class. If this() method is used when only one constructor is present within a class, then it would result in a recursive call and leads to StackOverFlow.
Eg:

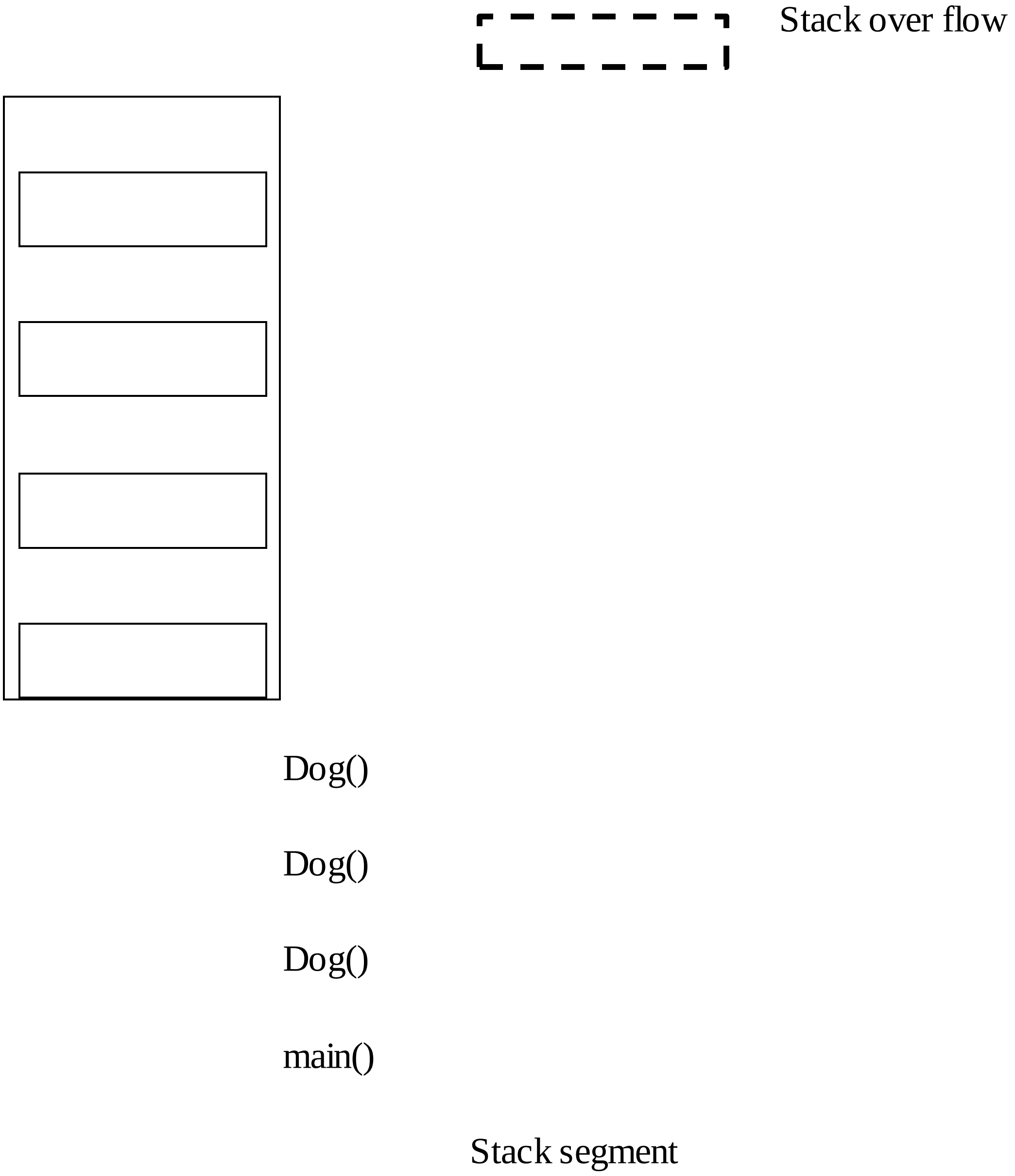
```
class Dog
{
        String name;
        String breed;
        int cost;

        Dog()
        {
                this();//error
        }
}

class Launch
{
        public static void main(String args[])
        {
                Dog d = new Dog();
        }
}
```

Output:

Error



22 Does every constructor call its superclass constructor?

Yes. If the programmer has not explicitly called this() method within a constructor then automatically super() method would be present and hence super class constructor gets called.

23 When is an implied super() not included in each constructor?

If the programmer manually calls this() method, then super() method would not be included.

24Can super() have parameters in it?

Yes. However if super() method calls object class constructor then it is not possible.

25 If there are multiple constructors in the superclass then which constructor does super() call?

Default constructor of the parent class would be called.

26Why is a constructor called a constructor?

It is the constructor that converts an empty object into a meaningful object by assigning values to the instance variables. JVM only allocates memory for an object. It is the constructor that constructs the object by giving values, so it is called constructor.

27What is a role of default constructor?

The default constructor within its body would have a call to the super class constructor. The super class constructor would initialize the instance variables to default values.

28What is the difference between constructor and ordinary methods?

Constructors	Ordinary methods
Has the same name as that of the class.	Does not have the same name as that of the class.
Has no return type	Has return type
It is called during object creation.	It is called after object creation.

It is involved in construction of an object.	It is involved in exhibiting other behaviours.
Does not participate in inheritance.	Participate in inheritance.
super() and this() methods can be used only inside a constructors.	super() and this() methods cannot be used within an ordinary methods.
Constructors cannot be abstract.	Ordinary methods can be abstract.
Cannot be final	Can be final
Cannot be static	Can be static
Cannot be native	Can be native
Cannot be synchronized	Can be synchronized.

29What are the operations that a constructor performs?

It normally performs initialization i.e construction of an object.

30 Can constructors be abstract?

No

31 Can constructors be final?

No

32Can constructors be native?

No

33 Can constructors be static?

No

34 Can constructors be synchronized?

No

35 Can constructors be public?

Yes

36 Can constructors be private?

Yes. If in case it is made private then it would enforce singleton design pattern.

37Can constructors be protected?

Yes

38 Can constructors be default?

Yes

39What is the difference in the use of “this” by a constructor and an ordinary method?

“this” in a constructor plays two role.

i) performs local chaining and

ii) resolves shadowing problem.

“this” keyword in an ordinary method is used to resolve shadowing problem. However this() cannot be used in an ordinary method.

40 What is the difference in the use of “super” by a constructor and an ordinary method?

In ordinary method “super” keyword is used to access a super class variable which has same name as that of the subclass variables. However super() cannot be used in an ordinary method.

In case of constructors “super” keyword is used to access a super class variable which has same name as that of the subclass variables and super() would be used for constructor chaining.

41 When we create a class such as public class Example{ } what extra additions would be performed by the compiler automatically?

```
public class Example extends Object
        {
                public Example()
                {
```

```
                Super();
        }
}
public class Object
{
        public Object()
        {
                ---
                ---
        }
}
```

42Can constructors be inherited?

No

43Can ordinary methods be inherited?

Yes

44How is constructor chaining achieved in java?

Using super() method.

45How are this() and super() used with constructors?

this() method is used for local chaining and super() method is for constructor chaining.

46What is the access modifier of the default constructor?

Default constructor would be having the same access modifier as that of the class to which it belongs to.

47What happens when a constructor is declared as private?

It results in singleton design pattern.

48 What is a constructor in java?

Constructor is a specialized setter which performs initialization of an instance variables that are present within an object.

49Why do we require overloaded constructors in java?

The objects that are created for a class may demand different types of initialization.

Some objects may have to be initialized to their default values in which case, default constructor is required.

Some objects may have to be initialized to different values as required by the programmer for which parameterized constructors must be used.

Therefore a class would normally contain overloaded constructor.

50Does a class inherit the constructors of its superclass?

No

51Can a top level class be private or protected?

It can't be both. It should be either public or default.

52What type of parameter passing does Java support?

pass by value and pass by reference.

53Primitive data types are passed by reference or pass by value?

pass by value.

54Objects are passed by value or by reference?

pass by reference.