

1) What is a class?

→ A java class is a definition block used to define the state and behaviors of an object.

→ A class contains members, members are of two types,
(i) data member
(ii) member function.

→ Data members are variables used to define the states of an object.

→ Member functions are methods used to define the behaviors of an object.

2) What is an object?

→ An object is a real world entity that has its own states and behaviors.

Syntax

className variable = new className();

→ In java, an object is a copy of the class that contains

all the non-static members.

→ The non-static members of a class gets loaded to memory during object creation.

3) What is a reference variable?

→ A variable that is declared using non-primitive datatype is called as a reference variable.

- Generally we declare reference variable using,
- (i) class name
 - (ii) interface name
 - (iii) enum name

→ In java, a reference variable can store either address of an object or null value.

4) What is null? When we get NullPointerException?

→ null is a keyword in java.

→ null meaning is nothing.

→ If a reference variable assigned null, it is called null reference.

→ null is the default value for all reference variable.

→ If we try to access any non-static members using null reference, it will cause NullPointerException.

5) D

→ M
of
ref
dist
dee

→ In
made
will

6) Diffe

- (i) creat
- (ii) acces
- (iii) load
- (iv) only
- (v) loaded
- (vi) loaded
- (vii) class

variable?

are using non-primitive reference variable.

reference variable using,

variable can store either null value.

get NullPointerException?

assigned null, it is called

use for cell reference variable.
non-static members
will cause "NullPointerException"

5) Difference between DEEP copies and SHALLOW copies?

DEEP COPIES

→ Multiple reference variable of a same class with each reference variable assigned to distinct object is called as deep copies.

→ In deep copies, changes made by one reference variable will not affect others.

SHALLOW COPIES

→ Multiple reference variable of a same class assigned to a common object is called shallow copies.

→ In shallow copies changes made by one reference variable will affect the others.

6) Difference between static and non-static Members?

Static Members

Non-Static Members

- (i) created with static keyword (ii) created without static keyword.
- (iii) accessed using class name (iv) accessed by creating an instance.
- (v) loaded to class area (vi) loaded to heap area.
- (vii) only 1 copy exist in memory (viii) multiple copies may exist in memory.
- (ix) loaded during class loading (x) loaded during instantiation.
- (xi) loaded by class loaders (xii) loaded by JVM.
- (xiii) class level members (xiv) instance members.

7) Difference b/w static block and non-static block?

Static Block

- => Created with static keyword.
- => Execute by class loader.
- => Execute during class loading.
- => Execute only once.

Non-static Block

- => Created without static keyword.
- => Execute by the JVM.
- => Execute during object creation.
- => Execute multiple times.

8) What is classLoader?

Ans) It is a sub-system of the JVM present inside the JRE.

=> It is primarily responsible for:

1. Loading all the static members to memory.
2. Executing all the static initialization blocks.

=> The time interval in which class loader works is known as class loading time.

9) Can we initialize non-static data members inside a static block?

Ans) No. We can't initialize static data members inside a static block.

10) We can initialize static data members inside a static block?

Ans) Yes, but it is not recommended.

11) What is constructor?

Ans) Constructors are used to:

=> Create objects.

=> Constructors have no return type.

Type

(i) Default

(ii) User-defined

b) z

b) p

12) Constructors

Ans) Constructors are used to:

=> Create objects.

=> Multiple constructors.

=> Rules:-

- When creating, mandatory to

- We can very

- (i) Types of

- (ii) Length of

- (iii) Order of

=> Uses

Constructor overloading the same class

ex:-

non-static block?

This block

without static

by the JVM.

using object creation.

multiple times.

able to initialize
data members.

VM present inside

is to memory
zation blocks.

class loader works

members inside a

instance

inside a

1) What is a constructor?

Ans)

⇒ Constructors are special members in a class, that are used to initialized non-static data members.

⇒ Constructor will always have same name as class name.

⇒ Constructors get executed during object creation.

Types

(i) Default constructor

(ii) User defined constructor

↳ zero argument constructor

↳ Parameterized constructor

ex:-

2) Constructors overloading

Ans)

⇒ constructor overloading is the concept of developing multiple constructor inside a class.

⇒ Rules:-

- When creating multiple constructor inside a class, it is mandatory to vary the argument list in each constructor.

- We can vary the arguments in 3 ways,

- (i) Types of arguments

- (ii) length of arguments

- (iii) Orders of arguments

⇒ Uses

constructor overloading is used to create object of the same class with different initializations.

ex:-

13) What is Inheritance?

⇒ Inheritance is the concept of a class acquiring the properties of another class.

⇒ Inheritance can be done using extends keyword.

⇒ The concept of inheritance is also known as IS-A Relationship.

Rules

⇒ The sub-class can't acquire the following members of the super-class.

(i) Private members

(ii) static members

(iii) Constructors

(iv) Initialization blocks.

⇒ The super-class can't access the properties of the sub-class.

⇒ When a class is declared as final, it can't have a sub-class.

Uses / Advantages

(i) code Reusability

(ii) S/W Enhancement

(iii) code Modification

(iv) Generalization & Specialization.

Types of Inheritance

(i) Single level inheritance

(ii) Multi level inheritance

(iii) Hierarchical inheritance

(iv) Multiple inheritance

(v) Hybrid inheritance.

14) Explain using

⇒ Multi

It lead

⇒ In mul

from c

properties

⇒ It also

⇒ The super
doesn't k

15) What is

Ans) ⇒ Generali
functionalitie

the purpos

in sub cla

16) Can const

Ans) No, if
behaved li

⇒ Can consta

Ans) No, but

18) Can we zni

constructor?

Ans) Yes, but it

19) Can we inher

Ans) No, but we

20) Can we inheri

if a class acquiring
extends keyword.
also known as
the following members

14) Explain why multiple inheritance is not supported using classes?

⇒ ~~multiple inheritance~~

It leads to Jacob's Diamond Ambiguity Problem.

⇒ In multiple inheritance, the sub-class doesn't know from which super-class it must acquire the properties of object class.

⇒ It also creates problems w.r.t. constructor chaining.

⇒ The super() statement of the sub-class constructor doesn't know which constructor should be called.

15) What is Generalization and specialization?

Ans) ⇒ Generalization is the process of developing common functionalities in super-class, and specialization is the process of developing specific functionalities in sub-class.

16) Can constructors have a return type?

Ans) No, if by mistake we provide return type it behaves like a method.

17) Can constructors have modifiers?

Ans) No, but we can provide access-modifiers.

18) Can we initialize static data members inside a constructor?

Ans) Yes, but it is not recommended.

19) Can we inherit static members?

Ans) No, but we can access.

20) Can we inherit final class? No.

Q1) Can a final class have super class?

A1) Yes.

Q2) What is constructor chaining?

A2) Constructor chaining is the phenomenon of one constructor calling another constructor.

(i) There are two ways to achieve constructor chaining:

(a) Using this() statement

(b) Using super() statement.

(ii) this() statement

⇒ this() statement is used to call another constructor of the same class.

⇒ We have to write this() statement in the first line of the constructor.

⇒ We can write only one this() statement in the constructor and it can be written by the programmer.

(iv) super() statement

⇒ super() statement can be used to call another constructor of the super class.

⇒ super() statement should always be written in the first line of the constructor.

⇒ We can write only one super() statement.

⇒ It can be written either by the programmer or by the compiler.

Q3) Difference between this() and super()

this

⇒ this()
call constructor
of current class.

⇒ should
by programmer

Class ?

23) Difference between this() and super() statements?

this() statement

⇒ this() statement is used to call constructor of same class.

⇒ should always be written by programmer.

super() statement

⇒ super() statement is used to call constructor of super class.

⇒ can be written by either programmer or compiler.