41. Do we require a parameter for constructors?

A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor

42. Explain static and dynamic binding?

Static binding:-

The binding which can be resolved at compile time by the compiler is known as static or early binding. The binding of all the static, private, and final methods is done at compile-time.

Dynamic binding:-

In Dynamic binding compiler doesn’t decide the method to be called. Overriding is a perfect example of dynamic binding. In overriding both parent and child classes have the same method.

43. How many instances can be created for an abstract class?

The answer to the question of how many instances of an abstract class can be created is zero. That is, we cannot create an instance of an abstract class as it does not have any complete implementation. An abstract class acts like a template or an empty structure.

44. Explain the default access specifiers in a class definition?

**Default** is a keyword that is used as an access modifier for methods and variables.  
Using this access modifier will make your class, variable, method or constructor acessible from own class or package, it will be also is set if no access modifier is present.

45. Which OOPS concept is used as reuse mechanism?

Inheritance is the feature that provides a reuse mechanism.

This mechanism provides reusability to the user. While abstraction, encapsulation, and dynamic binding have different functionalities in the OOP paradigm

46. Define the Benefits of Object Oriented Programming?

The Benefits of Object Oriented Programming are as follows:-

* Modularity for easier troubleshooting. ...
* Reuse of code through inheritance. ...
* Flexibility through polymorphism. ...
* Effective problem solving. ...
* What to know about OOP developer jobs.

47. Explain method overloading?

If a [class](https://www.javatpoint.com/object-and-class-in-java) has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

If we have to perform only one operation, having same name of the methods increases the readability of the [program](https://www.javatpoint.com/java-programs).

Advantage of method overloading is that it *increases the readability of the program*

48. Explain the difference among early binding and late binding?

|  |  |
| --- | --- |
| Early Binding | Late Binding |
| It is a compile-time process | It is a run-time process |
| The method definition and method call are linked during the compile time. | The method definition and method call are linked during the run time |
| Actual object is not used for binding. | Actual object is used for binding. |
| For example: Method overloading | For example: Method overriding |
| Program execution is faster | Program execution is slower |

49. Explain early binding? Give examples?

Early Binding:-

The binding which can be resolved at compile time by the compiler is known as static or early binding. Binding of all the static, private and final methods is done at compile-time.

For example: Method overloading

50. Explain loose coupling and tight coupling?

Tight Coupling:-

The tightly coupled object is an object that needs to know about other objects and is usually highly dependent on each other's interfaces.

Changing one object in a tightly coupled application often requires changes to a number of other objects.

In the small applications, we can easily identify the changes and there is less chance to miss anything. But in large applications, these inter-dependencies are not always known by every programmer and there is a chance of overlooking changes.

Loose Coupling

Loose coupling is a design goal to reduce the inter-dependencies between components of a system with the goal of reducing the risk that changes in one component will require changes in any other component.

Loose coupling is a much more generic concept intended to increase the flexibility of the system, make it more maintainable and makes the entire framework more stable.