**Interview Questions**

1. What is Variable?
2. What is Method?
3. What is Method Overloading?
4. What is Method Overriding?
5. What is type casting?
6. Explain Primitive and Non-Primitive.
7. What is Inheritance? Why multiple Inheritance is not possible through Java classes?
8. What is Polymorphism? Explain its types.
9. What is Interface and Abstract class?

10.What is the difference between static and non-static?

11.What is encapsulation? And what is java bean class?

12.what is exception handling? And how to handle it?

13.What is collection? Explain the collection classes with an example.

14.Write a program for pass by reference with a real time example.

15. What is composition?

16. Why java is called object oriented programming language?

17. What are the java features?

* It is compiled.
* It is Robust.
* It is highly secured language.
* It is simple and easy to understand.
* High performance.
* Multithreaded.
* It is platform independent.

18. What is the difference between methods and constructors?

|  |  |
| --- | --- |
| It is used to perform some operations. | It is used to initialize data members. |
| It will have return type. | It will not have any return type. |
| We can overload and override it. | We can overload but cannot override it. |
| It can be classified into static and non-static | It’s always non-static. |
| It can be inherited. | Cannot be inherited. |
| It can be any name even class name. | It should always be same as class name. |
| It will be called or invoked by name. | It will be invoked whenever an object is created. |
| It can return any value. | It doesn’t return any value. |
| It should be developed explicitly. | Each and every class will have a constructor. |

19.What is the difference between this keyword and super keyword?

|  |  |
| --- | --- |
| THIS keyword is used to point to the current object. | SUPER keyword is used in the case of method overriding along with the subclass implementation, if we need super class implementation then we should go for SUPER keyword. |
| Whenever a global and local variable names are same then we should go for THIS keyword. | In the case of method overriding we use SUPER keyword. |
| THIS keyword can be used only in the non-static context because internally THIS keyword is non-static. | SUPER keyword can be used only in non-static context. |

20. Which keyword is the default reference variable in java?

-> THIS keyword.

21. What is difference method overloading and method overriding?

|  |  |
| --- | --- |
| Developing multiple methods with same name but variation in the argument list is called as method overloading. | Developing a sub class with same name and signature as in the super class but with different implementation in the sub class is called method overriding. |
| We can overload bot static and non-static. | We can override ony non-static. |
| The method name should be same but there is no restrictions on the access specifier, modifier and return type. | The complete signature should be same in the sub class as in the super class. |
| It is used to perform some operations. | When we want new implementation we go for method overriding. |
| There is no rule that it should have IS A relationship. | It should have IS A relationship to achieve overriding. |
| It is an example for compile time polymorphism | It is an example for run time polymorphism. |
| We can overload main method | We cannot override main method. |

22. What is the difference between this calling and super calling?

23. Why main method is declared as public static void main(String[] args) ?

-🡪 public is application level access where JVM should be able to access the main method hence it is declared as public.

It is static because it should be loaded into the memory before execution and it should be one copy.

Return type is void because it doesn’t return any value.

Min is method name.

The parameters are string[] argsbecause it receives input in the staring[] format at the command line.

Command prompt.

Class Demo

{

Public static void main(String[] args)

{

String[] arr={“Hi”,”Hello”};

For(int i=0;i<arr.length;i++)

{

S.O.P(“arr[i];

}

}

}

Compilation: javac Demo.java

Interpretation: java Demo “$97a” /\*input is given in command promt\*/

Eclipse

Package collectiontopic;

Public class sample

{

P.S.V.M(\_\_\_)

{

For (int i=0;i<args.length;i++)

{

S.O.P(“args[i]);

}

}

}//Run->Run confir->arguments we should pass.

24. What is the difference between abstract class and Interface?

|  |  |
| --- | --- |
| Any class which is declared with a keyword abstract is called abstract class. | Interface is a java type which is by default abstract. |
| In abstract class we have 3 members. | In interface we have 2 members. |
| In abstract class we have constructor | We don’t have constructor. |
| Variables in abstract class can be static and non-static. | Variables in interface is by default static and final. |
| Methods can have access specifier. | Methods are by default public and abstract. |
| We can develop both abstract and concrete methods. | We can develop only abstract methods. |
| We have to override all the abstract methods in the sub class but not in the concrete class. | All the methods should be overridden. |
| Abstarct class extends object class. | Interface doesn’t extend any class. |
| Through abstract class we can achieve constructor chain. | Through interface we c cannot achieve constructor chain because it doesn’t support constructor |
| When we know partial implementation then we should go for abstraction. | When we don’t know % implementation the we should go for interface. |

25. What is constructor chaining?

🡪 A subclass constructor calling its immediate super class constructor, super class constructor calling its immediate super class constructor is called as constructor chaining.

26. What is the difference error and exception?

|  |  |
| --- | --- |
| Error cannot be predicted. | Exception can be predicted. |
| Errors are occurred due to the system configuration. | Exceptions are occurred due to the mistakes done by the programmer. |
| Error cannot be handled. | Exception can be handled. |

27. Write a program to reverse a string

i/p : JAVA MAVA JOBAA

o/p : JOBAA MAVA JAVA

class Demo

{

P.S.V.M(\_\_\_\_\_)

{  
 string s1=”JAVA MAVA JOBAA”;

String [] arr=s1.split(“”);

For(int i=arr.length-1;i>=0;i++)

{

S.O.P(arr[i]);

}

}

}