Q1 What is a class?

Class is a blue print from which objects are created. It does not occupy memory in the heap.

Q2. What is an object?

Object is an instance of a class which occupies memory in the heap area.

Q3. What is abstraction in Java?

Abstraction means hiding the implementation

Q4. How will you achieve abstraction?

Using abstract class or interface.

Q5. What is abstract class?

Abstract class is a restricted class that cannot be instantiated. To use abstract class it must be inherited. Abstract class can have both abstract as well as non-abstract methods.

Q5. What is interface?

Interface is an abstract type which cannot be instantiated. Interface can have only abstract methods and constant declarations (pi=3.14). **Since java8 an interface can have static and default methods also.**

Q5A What is the advantage of interface in java?

To achieve abstraction, to achieve multiple inheritance, to achieve loose coupling.

Q6. Difference between abstract class and interface

Abstract class can have both abstract and non-abstract methods.

Interface can have only abstract methods before java8.

Q7. What is inheritance?

Inheritance is a technique by which a child class can access the properties and methods of its parent class. Code re-usability is the advantage.

Q8. What is encapsulation or **data hiding**?

Declaring the data members of a class as private is encapsulation or data hiding.

Q9. What is polymorphism in OOP?

Same object behaving differently in different scenarios. Method overloading and Method overriding are the two types of polymorphism.

Q10. What is method overloading/early binding/static polymorphism/compile time polymorphism?

Methods with same name having different parameters.

Linking method call and method definition is binding

Q11. What is method overriding?

Method overriding means having the same method in the child class and the parent class.

Q12. Java doesn’t support multiple inheritance in java, why?

Multiple inheritance means having more than one parent class.

To avoid ambiguity as to which parent method is to be overridden when both the parents having the same method.

Q13. What is a constructor?

A constructor is a bock to initialize the properties.

A constructor is called whenever an object is created using the new key word.

A constructor does not have a return type because it returns nothing (not even void)

Java compiler will provide a default constructor (no argument constructor) if there is no parameterized constructor.

**Part 2**

Q14. Can you override a static method?

No.

Static methods are class level methods. Therefore binding happens during compile time. Method overriding takes place during run time.

Q15. **Exception**

Exception is an abnormal condition which occurs during the run time and disrupts normal flow of a program.

Exception is the super class of all Exceptions.

Q16. How do you handle exceptions?

Using try, catch blocks.

try block : code is put in the try block

catch block : catches the exceptions occurred in the try block.

finally block: finally block is always executed whether exception is occurred or not.

Q17. Error Vs Exception

Both Error and Exception are classes in the java.lang package.

Error occurs due to the shortage of system resources. OutofMemoryError, StackOverflowError, ClassDefNotFoundError etc are examples.

Exceptions occurs from within the application, which are supposed to be handled. ArrayIndexOutOfBoundException, NullPointerException, FileNotfoundException, ClassNotFoundException etc.

Q18. What does a return statement do?

A return statement returns the control.

Q19. What is Checked and Unchecked exception?

Checked exception is the exception that is known to the compiler. Eg:– FileNotFoundException, IOException, ClassNotFoundException

Unchecked exception is the exception that the compiler is not aware of. Eg:- NullPointerException, ArrayIndexOutOfBoundException, ArithmeticException

Q20.

|  |  |
| --- | --- |
| **ClassDefNotFoundError** | **ClassNotFoundException** |
| It is an error. It is of type java.lang.Error | It is an exception. It is of type java.lang.Exception |
| It occurs when Java runtime system doesn’t find a class definition, which is present at compile time, but missing at run time. | It occurs when an application tries to load a class at run time which is not updated in the class path. |

Q21. What is the purpose of finally block?

Finally block is used to keep clean up operations like closing DB resources.

**Difference between Checked and Unchecked Exception**

|  |  |  |
| --- | --- | --- |
|  | **Checked Exception** | **Unchecked Exception** |
| Compile time checking | Yes | No |
| Handling of exception | Must be handled by either using try/catch block or throws keyword | Not necessary to handle |
| Examples | SQLException, IOException, FileNotFoundException | ArithmeticException, ArrayIndexOutOfBoundException,  NullPointerException |

Q22. Difference between **throw** and **throws** in java?

‘**throw**’ is a statement used **in a method** to explicitly throw an exception. The ‘throw’ statement can be used to throw only one exception in a single statement. Eg:- throw new ArthmeticException

‘**throws**’ is a keyword used in the **method definition** to specify which exceptions can be thrown from the method. The ‘throws’ key word can be used to throw multiple exceptions separated by a comma. Eg:- throws ArthmeticException, NullPointerException

Q23. **Does finally block always execute in Java?**

No, there is one scenario where the finally block does not execute. When you run System.exit(0) in the try or catch block, then finally block does not execute.

Q24 Difference between **try/catch** and **throws**

|  |  |
| --- | --- |
| [**try-catch block**](https://www.benchresources.net/try-catch-block-in-java-exception-handling/) | [**throws keyword**](https://www.benchresources.net/throws-keyword-in-java-exception-handling/) |
| Using try-catch block, we can handle exception. | Whereas using throws keyword, we can simply declare exception that might raise from that method |
| A user friendly message can be given to the user | In most cases, throws declaration leads to abnormal termination |
| Caught exception in the catch block can be re-thrown after some alteration | There is no such flexibility, as its directly throws exception |

Q25. What is try-with-resources statement?

While using try/catch block, finally block is used to close any open resources.

Try-with-resources was introduced in 1.7 version for automatic resource management. Programmer doesn’t need to explicitly close opened resources; rather it will be automatically closed once control reaches the end of try-catch block.

|  |
| --- |
| public static void main(String[] args) {            try (FileReader obj =  new FileReader("test.txt")) {                System.out.println(obj.readLine());          }          catch (IOException e) {                System.out.println(e);        }      } |

In the above example we don’t have to include a finally block to close the FileReader object. It will be closed automatically when the control leaves the catch block.