

69

CORE JAVA

INTERVIEW QUESTIONS

YOU'LL MOST LIKELY BE ASKED

CREATE BY - ATUL KUMAR (LINKEDIN)

1) what are static blocks and static initializers in Java ?

Static blocks or static initializers are used to initialize static fields in java. we declare static blocks when we want to initialize static fields in our class. Static blocks gets executed exactly once when the class is loaded . Static blocks are executed even before the constructors are executed.

2) How to call one constructor from the other constructor ?

With in the same class if we want to call one constructor from other we use this() method. Based on the number of parameters we pass appropriate this() method is called.

Restrictions for using this method :

- 1) this must be the first statement in the constructor
- 2) we cannot use two this() methods in the constructor

3) What is method overriding in java ?

If we have methods with same signature (same name, same signature, same return type) in super class and subclass then we say subclass method is overridden by superclass.

When to use overriding in java

If we want same method with different behaviour in superclass and subclass then we go for overriding. When we call overridden method with subclass reference subclass method is called hiding the superclass method.

4) What is super keyword in java ?

Variables and methods of super class can be overridden in subclass . In case of overriding , a subclass object call its own variables and methods. Subclass cannot access the variables and methods of superclass because the overridden variables or methods hides the methods and variables of super class. But still [java](#) provides a way to access super class members even if its members are overridden. Super is used to access superclass variables, methods, constructors.

Super can be used in two forms :

- 1) First form is for calling super class constructor.
 - 2) Second one is to call super class variables, methods.
- Super if present must be the first statement.

5) Difference between method overloading and method overriding in java ?

Method Overloading	Method Overriding
1) Method Overloading occurs with in the same class	Method Overriding occurs between two classes superclass and subclass
2) Since it involves with only one class inheritance is not involved.	Since method overriding occurs between superclass and subclass inheritance is involved.
3) In overloading return type need not be the same	3) In overriding return type must be same.
4) Parameters must be different when we do overloading	4) Parameters must be same.
5) Static polymorphism can be achieved using method overloading	5) Dynamic polymorphism can be achieved using method overriding.
6) In overloading one method can't hide the another	6) In overriding subclass method hides that of the superclass method.

6) Difference between abstract class and interface ?

Interface	Abstract Class
1) Interface contains only abstract methods	1) Abstract class can contain abstract methods, concrete methods or both
2) Access Specifiers for methods in interface must be public	2) Except private we can have any access specifier for methods in abstract class.
3) Variables defined must be public , static , final	3) Except private variables can have any access specifiers
4) Multiple Inheritance in java is implemented using interface	4)We cannot achieve multiple inheritance using abstract class.
5) To implement an interface we use implements keyword	5)To implement an interface we use implements keyword

7) Why java is platform independent?

The most unique feature of java is platform independent. In any programming language source code is compiled in to executable code . This cannot be run across all platforms. When javac compiles a java program it generates an executable file called .class file.

class file contains byte codes. Byte codes are interpreted only by JVM's . Since these JVM's are made available across all platforms by Sun Microsystems, we can execute this byte code in any platform. Byte code generated in windows environment can also be executed in linux environment. This makes java platform independent.

8) What is method overloading in java ?

A class having two or more methods with same name but with different arguments then we say that those methods are overloaded. Static polymorphism is achieved in java using method overloading.

Method overloading is used when we want the methods to perform similar tasks but with different inputs or values. When an overloaded method is invoked java first checks the method name, and the number of arguments ,type of arguments; based on this compiler executes this method.

Compiler decides which method to call at compile time. By using overloading static polymorphism or static binding can be achieved in java.

Note : Return type is not part of method signature. we may have methods with different return types but return type alone is not sufficient to call a method in java.

9) What is difference between c++ and Java ?

Java	C++
1) Java is platform independent	C++ is platform dependent.
2) There are no pointers in java	There are pointers in C++.
3) There is no operator overloading in java	C ++ has operator overloading.
4) There is garbage collection in java	There is no garbage collection
5) Supports multithreading	Does'nt support multithreading
6) There are no templates in java	There are templates in java
7) There are no global variables in java	There are global variables in c++

10) What is JIT compiler ?

JIT compiler stands for Just in time compiler. JIT compiler compiles byte code in to executable code . JIT a part of JVM .JIT cannot convert complete java program in to executable code it converts as and when it is needed during execution.

11) What is bytecode in java ?

When a javac compiler compiles a class it generates .class file. This .class file contains set of instructions called byte code. Byte code is a machine independent language and contains set of instructions which are to be executed only by JVM. JVM can understand this byte codes.

12) Difference between this() and super() in java ?

this() is used to access one constructor from another with in the same class while super() is used to access superclass constructor. Either this() or super() exists it must be the first statement in the constructor.

13) What is a class ?

Classes are fundamental or basic unit in Object Oriented Programming .A class is kind of blueprint or template for objects. Class defines variables, methods. A class tells what type of objects we are creating. For example take Department class tells us we can create department type objects. We can create any number of department objects.

All programming constructs in [java](#) reside in class. When JVM starts running it first looks for the class when we compile. Every Java application must have atleast one class and one main method. Class starts with class keyword. A class definition must be saved in class file that has same as class name. File name must end with .java extension.

```
public class FirstClass
{public static void main(String[] args)
{System.out.println("My First class");
}
}
```

If we see the above class when we compile JVM loads the FirstClass and generates a .class file(FirstClass.class). When we run the program we are running the class and then executes the main method.

14) What is an object ?

An Object is instance of class. A class defines type of object. Each object belongs to some class. Every object contains state and behavior. State is determined by value of attributes and behavior is called method. Objects are also called as an instance.

To instantiate the class we declare with the class type.

```
public class FirstClass {public static void main(String[] args)
{
FirstClass f=new FirstClass();
System.out.println("My First class");
}
}
```

To instantiate the FirstClass we use this statement
FirstClass f=new FirstClass();
f is used to refer FirstClass object.

15)What is method in java ?

It contains the executable body that can be applied to the specific object of the class.

Method includes method name, parameters or arguments and return type and a body of executable code.

```
Syntax : type methodName(Argument List){
}
```

ex : public float add(int a, int b, int c)

methods can have multiple arguments. Separate with commas when we have multiple arguments.

16) What is encapsulation ?

The process of wrapping or putting up of data in to a single unit class and keeps data safe from misuse is called encapsulation .

Through encapsulation we can hide and protect the data stored in [java](#) objects. Java supports encapsulation through access control. There are four access control modifiers in java public , private ,protected and default level.

For example take a car class , In car we have many parts which is not required for driver to know what all it consists inside. He is required to know only about how to start and stop the car. So we can expose what all are required and hide the rest by using encapsulation.

17) Why main() method is public, static and void in java ?

public : "public" is an access specifier which can be used outside the class. When main method is declared public it means it can be used outside class.

static : To call a method we require object. Sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.

void : void return type is used when a method doesn't return any value . main() method doesn't return any value, so main() is declared as void.

Signature : public static void main(String[] args) {

18) Explain about main() method in java ?

Main() method is starting point of execution for all java applications.

public static void main(String[] args) {}

String args[] are array of string objects we need to pass from command line arguments. Every Java application must have atleast one main method.

19)What is constructor in java ?

A constructor is a special method used to initialize objects in java.

we use constructors to initialize all variables in the class when an object is created. As and when an object is created it is initialized automatically with the help of constructor in java.

We have two types of constructors

Default Constructor

Parameterized Constructor

Signature : public classname()

```
{  
}
```

Signature : public classname(parameters list)

```
{  
}
```

20) What is difference between length and length() method in java ?

length() : In String class we have length() method which is used to return the number of characters in string.

Ex : String str = "Hello World";

System.out.println(str.length());

Str.length() will return 11 characters including space.

length : we have length instance variable in arrays which will return the number of values or objects in array.

For example :

String days[]={ " Sun", "Mon", "wed", "thu", "fri", "sat"};

Will return 6 since the number of values in days array is 6.

21) What is ASCII Code?

ASCII stands for American Standard code for Information Interchange. ASCII character range is 0 to 255. We can't add more characters to the ASCII Character set. ASCII character set supports only English. That

is the reason, if we see C language we can write C language only in English we can't write in other languages because it uses ASCII code.

22) What is Unicode ?

Unicode is a character set developed by Unicode Consortium. To support all languages in the world [Java](#) supports Unicode values. Unicode characters were represented by 16 bits and its character range is 0-65,535.

Java uses ASCII code for all input elements except for Strings, identifiers, and comments. If we want to use Telugu we can use Telugu characters for identifiers. We can enter comments in Telugu.

23) Difference between Character Constant and String Constant in java ?

Character constant is enclosed in single quotes. String constants are enclosed in double quotes. Character constants are single digit or character. String constants are collection of characters.

Ex : '2', 'A'

Ex : "Hello World"

24) What are constants and how to create constants in java?

Constants are fixed values whose values cannot be changed during the execution of program. We create constants in Java using final keyword.

Ex : final int number = 10;

final String str = "java-interview -questions"

25) Difference between '>>' and '>>>' operators in java?

>> is a right shift operator shifts all of the bits in a value to the right to a specified number of times.

int a = 15;

a = a >> 3;

The above line of code moves 15 three characters right.

>>> is an unsigned shift operator used to shift right. The places which were vacated by shift are filled with zeroes.

Core Java Interview questions on Coding Standards

26) Explain Java Coding Standards for classes or Java coding conventions for classes?

Sun has created Java Coding standards or Java Coding Conventions. It is recommended highly to follow Java coding standards.

Classnames should start with uppercase letter. Classnames names should be nouns. If Class name is of multiple words then the first letter of inner word must be capital letter.

Ex : Employee, EmployeeDetails, ArrayList, TreeSet, HashSet

27) Explain Java Coding standards for interfaces?

1) Interface should start with uppercase letters

2) Interfaces names should be adjectives

Example : Runnable, Serializable, Marker, Cloneable

28) Explain Java Coding standards for Methods?

1) Method names should start with small letters.

2) Method names are usually verbs

3) If method contains multiple words, every inner word should start with uppercase letter.

Ex : toString()

4) Method name must be combination of verb and noun

Ex : getCarName(), getCarNumber()

29) Explain Java Coding Standards for variables ?

1) Variable names should start with small letters.

2) Variable names should be nouns

3) Short meaningful names are recommended.

4) If there are multiple words every inner word should start with Uppercase character.

Ex : string, value, empName, empSalary

30) Explain Java Coding Standards for Constants?

Constants in Java are created using static and final keywords.

1) Constants contains only uppercase letters.

2) If constant name is combination of two words it should be separated by underscore.

3) Constant names are usually nouns.

Ex: MAX_VALUE, MIN_VALUE, MAX_PRIORITY, MIN_PRIORITY

31) Difference between overriding and overloading in java?

Overriding	Overloading
In overriding method names must be same	In overloading method names must be same
Argument List must be same	Argument list must be different atleast order of arguments.
Return type can be same or we can return covariant type. From 1.5 covariant types are allowed	Return type can be different in overloading.
We cant increase the level of checked exceptions. No restrictions for unchecked exceptions	In overloading different exceptions can be thrown.
A method can only be overridden in subclass	A method can be overloaded in same class or subclass
Private,static and final variables cannot be overridden.	Private , static and final variables can be overloaded.
In overriding which method is called is decided at runtime based on the type of object referenced at run time	In overloading which method to call is decided at compile time based on reference type.
Overriding is also known as Runtime polymorphism, dynamic polymorphism or late binding	Overloading is also known as Compile time polymorphism, static polymorphism or early binding.

32) What is 'IS-A' relationship in java?

'is a' relationship is also known as inheritance. We can implement 'is a' relationship or inheritance in [java](#) using extends keyword. The advantage of inheritance or is a relationship is reusability of code instead of duplicating the code.

Ex : Motor cycle is a vehicle

Car is a vehicle Both car and motorcycle extends vehicle.

33) What is 'HAS A' relationship in java?

'Has a' relationship is also known as "composition or Aggregation". As in inheritance we have 'extends' keyword we don't have any keyword to implement 'Has a' relationship in java. The main advantage of 'Has-A' relationship in java code reusability.

34) Difference between 'IS-A' and 'HAS-A' relationship in java?

IS-A relationship	HAS- A RELATIONSHIP
Is a relationship also known as inheritance	Has a relationship also known as composition or aggregation.
For IS-A relationship we uses extends keyword	For Has a relationship we use new keyword
Ex : Car is a vehicle.	Ex : Car has an engine. We cannot say Car is an engine
The main advantage of inheritance is reusability of code	The main advantage of has a relationship is reusability of code.

35) Explain about instanceof operator in java?

Instanceof operator is used to test the object is of which type.

Syntax : <reference expression> instanceof <destination type>

Instanceof returns true if reference expression is subtype of destination type.

Instanceof returns false if reference expression is null.

```
Example : public classInstanceOfExample {public static voidmain(String[] args) {Integer a =
newInteger(5);if (a instanceof java.lang.Integer) {
System.out.println(true);
} else {
System.out.println(false);
}
}
```

```
}  
  
}
```

Since a is integer object it returns true.

There will be a compile time check whether reference expression is subtype of destination type. If it is not a subtype then compile time error will be shown as Incompatible types

36) What does null mean in java?

When a reference variable doesn't point to any value it is assigned null.

Example : Employee employee;

In the above example employee object is not instantiate so it is pointed no where

37) Can we have multiple classes in single file ?

Yes we can have multiple classes in single file but it people rarely do that and not recommended. We can have multiple classes in File but only one class can be made public. If we try to make two classes in File public we get following compilation error.

"The public type must be defined in its own file".

38) What all access modifiers are allowed for top class ?

For top level class only two access modifiers are allowed. public and default. If a class is declared as public it is visible everywhere.

If a class is declared default it is visible only in same package.

If we try to give private and protected as access modifier to class we get the below compilation error.

Illegal Modifier for the class only public,abstract and final are permitted.

39) What are packages in java?

Package is a mechanism to group related classes ,interfaces and enums in to a single module.

Package can be declared using the following statement :

Syntax : package <package-name>

Coding Convention : package name should be declared in small letters.

package statement defines the namespace.

The main use of package is

- 1) To resolve naming conflicts
- 2) For visibility control : We can define classes and interfaces that are not accessible outside the class.

40) Can we have more than one package statement in source file ?

We can't have more than one package statement in source file. In any [java](#) program there can be atmost only 1 package statement. We will get compilation error if we have more than one package statement in source file.

41) Can we define package statement after import statement in java?

We can't define package statement after import statement in java. package statement must be the first statement in source file. We can have comments before the package statement.

42) What are identifiers in java?

Identifiers are names in [java](#) program. Identifiers can be class name, method name or variable name.

Rules for defining identifiers in java:

- 1) Identifiers must start with letter,Underscore or dollar(\$) sign.
- 2) Identifiers can't start with numbers .
- 3) There is no limit on number of characters in identifier but not recommended to have more than 15 characters
- 4) Java identifiers are case sensitive.
- 5) First letter can be alphabet ,or underscore and dollar sign. From second letter we can have numbers .
- 6) We should'nt use reserve words for identifiers in java.

43) What are access modifiers in java?

The important feature of encapsulation is access control. By preventing access control we can misuse of class, methods and members.

A class, method or variable can be accessed is determined by the access modifier. There are three types of access modifiers in java. public,private,protected. If no access modifier is specified then it has a default access.

44) What is the difference between access specifiers and access modifiers in java?

In C++ we have access specifiers as public,private,protected and default and access modifiers as static, final. But there is no such division of access specifiers and access modifiers in java. In Java we have access modifiers and non access modifiers.

Access Modifiers : public, private, protected, default

Non Access Modifiers : abstract, final, stricfp.

45) What access modifiers can be used for class ?

We can use only two access modifiers for class public and default.

public: A class with public modifier can be visible

- 1) In the same class
- 2) In the same package subclass
- 3) In the same package nonsubclass
- 4) In the different package subclass
- 5) In the different package non subclass.

default : A class with default modifier can be accessed

- 1) In the same class
- 2) In the same package subclass
- 3) In the same package nonsubclass
- 4) In the different package subclass
- 5) In the different package non subclass.

46) Explain what access modifiers can be used for methods?

We can use all access modifiers public, private,protected and default for methods.

public : When a method is declared as public it can be accessed

- 6) In the same class
- 7) In the same package subclass
- 8) In the same package nonsubclass
- 9) In the different package subclass
- 10) In the different package non subclass.

default : When a method is declared as default, we can access that method in

- 1) In the same class
- 2) In the same package subclass
- 3) In the same package non subclass

We cannot access default access method in

- 1) Different package subclass
- 2) Different package non subclass.

protected : When a method is declared as protected it can be accessed

- 1) With in the same class
- 2) With in the same package subclass
- 3) With in the same package non subclass
- 4) With in different package subclass

It cannot be accessed non subclass in different package.

private : When a method is declared as private it can be accessed only in that class.

It cannot be accessed in

- 1) Same package subclass
- 2) Same package non subclass
- 3) Different package subclass
- 4) Different package non subclass.

47) Explain what access modifiers can be used for variables?

We can use all access modifiers public, private,protected and default for variables.

public : When a variables is declared as public it can be accessed

- 1) In the same class

- 2) In the same package subclass
- 3) In the same package nonsubclass
- 4) In the different package subclass
- 5) In the different package non subclass.

default : When a variables is declared as default, we can access that method in

- 1) In the same class
 - 2) In the same package subclass
 - 3) In the same package non subclass
- We cannot access default access variables in
- 4) Different package subclass
 - 5) Different package non subclass.

protected : When a variables is declared as protected it can be accessed

- 1) With in the same class
 - 2) With in the same package subclass
 - 3) With in the same package non subclass
 - 4) With in different package subclass
- It cannot be accessed non subclass in different package.

private : When a variables is declared as private it can be accessed only in that class.

It cannot be accessed in

- 1) Same package subclass
- 2) Same package non subclass
- 3) Different package subclass
- 4) Different package non subclass.

48) What is final access modifier in java?

final access modifier can be used for class, method and variables. The main advantage of final access modifier is security no one can modify our classes, variables and methods. The main disadvantage of final access modifier is we cannot implement oops concepts in java. Ex : Inheritance, polymorphism.

final class : A final class cannot be extended or subclassed. We are preventing inheritance by marking a class as final. But we can still access the methods of this class by composition. Ex: String class

final methods: Method overriding is one of the important features in java. But there are situations where we may not want to use this feature. Then we declared method as final which will print overriding. To allow a method from being overridden we use final access modifier for methods.

final variables : If a variable is declared as final ,it behaves like a constant . We cannot modify the value of final variable. Any attempt to modify the final variable results in compilation error. The error is as follows

"final variable cannot be assigned."

49) Explain about abstract classes in java?

Sometimes we may come across a situation where we cannot provide implementation to all the methods in a class. We want to leave the implementation to a class that extends it. In such case we declare a class as abstract. To make a class abstract we use key word abstract. Any class that contains one or more abstract methods is declared as abstract. If we don't declare class as abstract which contains abstract methods we get compile time error. We get the following error.

"The type <class-name> must be an abstract class to define abstract methods."

Signature ; abstract class <class-name>

```
{
}
```

For example if we take a vehicle class we cannot provide implementation to it because there may be two wheelers , four wheelers etc. At that moment we make vehicle class abstract. All the common features of vehicles are declared as abstract methods in vehicle class. Any class which extends vehicle will provide its method implementation. It's the responsibility of subclass to provide implementation.

The important features of abstract classes are :

- 1) Abstract classes cannot be instantiated.
- 2) An abstract classes contains abstract methods, concrete methods or both.
- 3) Any class which extends abstract class must override all methods of abstract class.
- 4) An abstract class can contain either 0 or more abstract methods.

Though we cannot instantiate abstract classes we can create object references . Through superclass references we can point to subclass.

50) Can we create constructor in abstract class ?

We can create constructor in abstract class , it doesn't give any compilation error. But when we cannot instantiate class there is no use in creating a constructor for abstract class.

51) What are abstract methods in java?

An abstract method is the method which doesn't have any body. Abstract method is declared with keyword abstract and semicolon in place of method body.

Signature : public abstract void <method name>();

Ex : public abstract void getDetails();

It is the responsibility of subclass to provide implementation to abstract method defined in abstract class.

Java Exception Handling Interview questions

52) What is an exception in java?

In [java](#) exception is an object. Exceptions are created when an abnormal situations are arisen in our program. Exceptions can be created by JVM or by our application code. All Exception classes are defined in java.lang. In other words we can say Exception as run time error.

53) State some situations where exceptions may arise in java?

- 1) Accessing an element that does not exist in array.
- 2) Invalid conversion of number to string and string to number.
(NumberFormatException)
- 3) Invalid casting of class
(Class cast Exception)
- 4) Trying to create object for interface or abstract class
(Instantiation Exception)

54) What is Exception handling in java?

Exception handling is a mechanism what to do when some abnormal situation arises in program. When an exception is raised in program it leads to termination of program when it is not handled properly. The significance of exception handling comes here in order not to terminate a program abruptly and to continue with the rest of program normally. This can be done with help of Exception handling.

55) What is an error in Java?

Error is the subclass of Throwable class in java. When errors are caused by our program we call that as Exception, but some times exceptions are caused due to some environment issues such as running out of memory. In such cases we can't handle the exceptions. Exceptions which cannot be recovered are called as errors in java.

Ex : Out of memory issues.

56) What are advantages of Exception handling in java?

- 1) Separating normal code from exception handling code to avoid abnormal termination of program.
- 2) Categorizing in to different types of Exceptions so that rather than handling all exceptions with Exception root class we can handle with specific exceptions. It is recommended to handle exceptions with specific Exception instead of handling with Exception root class.
- 3) Call stack mechanism : If a method throws an exception and it is not handled immediately, then that exception is propagated or thrown to the caller of that method. This propagation continues till it finds an appropriate exception handler ,if it finds handler it would be handled otherwise program terminates abruptly.

57) In how many ways we can do exception handling in java?

We can handle exceptions in either of the two ways :

- 1) By specifying try catch block where we can catch the exception.
- 2) Declaring a method with throws clause .

58) List out five keywords related to Exception handling ?

- 1) Try
- 2) Catch
- 3) throw
- 4) throws
- 5) finally

59) Explain try and catch keywords in java?

In try block we define all exception causing code. In [java](#) try and catch forms a unit. A catch block catches the exception thrown by preceding try block. Catch block cannot catch an exception thrown by another try block. If there is no exception causing code in our program or exception is not raised in our code jvm ignores the try catch block.

Syntax :

```
try
{
}
Catch(Exception e)
{
}
```

60) Can we have try block without catch block?

Each try block requires atleast one catch block or finally block. A try block without catch or finally will result in compiler error. We can skip either of catch or finally block but not both.

61) Can we have multiple catch block for a try block?

In some cases our code may throw more than one exception. In such case we can specify two or more catch clauses, each catch handling different type of exception. When an exception is thrown jvm checks each catch statement in order and the first one which matches the type of exception is execution and remaining catch blocks are skipped.

Try with multiple catch blocks is highly recommended in java.

If try with multiple catch blocks are present the order of catch blocks is very important and the order should be from child to parent.

62) Explain importance of finally block in java?

Finally block is used for cleaning up of resources such as closing connections, sockets etc. if try block executes with no exceptions then finally is called after try block without executing catch block. If there is exception thrown in try block finally block executes immediately after catch block.

If an exception is thrown,finally block will be executed even if the no catch block handles the exception.

63) Can we have any code between try and catch blocks?

We shouldn't declare any code between try and catch block. Catch block should immediately start after try block.

```
try{
//code
}
System.out.println("one line of code"); // illegal
catch(Exception e){
//
}
```

64) Can we have any code between try and finally blocks?

We shouldn't declare any code between try and finally block. finally block should immediately start after catch block.If there is no catch block it should immediately start after try block.

```
try{
//code
}
System.out.println("one line of code"); // illegal
finally{
//
}
```

65) Can we catch more than one exception in single catch block?

From [Java 7](#), we can catch more than one exception with single catch block. This type of handling reduces the code duplication.

Note : When we catch more than one exception in single catch block , catch parameter is implicitly final.

We cannot assign any value to catch parameter.

Ex : `catch(ArrayIndexOutOfBoundsException || ArithmeticException e)`

```
{
```

}

In the above example e is final we cannot assign any value or modify e in catch statement.

66) What are checked Exceptions?

- 1) All the subclasses of Throwable class except error, Runtime Exception and its subclasses are checked exceptions.
- 2) Checked exception should be thrown with keyword throws or should be provided try catch block, else the program would not compile. We do get compilation error.

Examples :

- 1) IOException,
- 2) SQLException,
- 3) FileNotFoundException,
- 4) InvocationTargetException,
- 5) CloneNotSupportedException
- 6) ClassNotFoundException
- 7) InstantiationException

67) What are unchecked exceptions in java?

All subclasses of RuntimeException are called unchecked exceptions. These are unchecked exceptions because compiler does not checks if a method handles or throws exceptions.

Program compiles even if we do not catch the exception or throws the exception.

If an exception occurs in the program, program terminates . It is difficult to handle these exceptions because there may be many places causing exceptions.

Example : 1) Arithmetic Exception

- 3) ArrayIndexOutOfBoundsException
- 4) ClassCastException
- 5) IndexOutOfBoundsException
- 6) NullPointerException
- 7) NumberFormatException
- 8) StringIndexOutOfBoundsException
- 9) UnsupportedOperationException

68) Explain differences between checked and Unchecked exceptions in java?

Unchecked Exception	Checked Exception
1) All the subclasses of RuntimeException are called unchecked exception.	All subclasses of Throwable class except RuntimeException are called as checked exceptions
2) Unchecked exceptions need not be handled at compile time	Checked Exceptions need to be handled at compile time.
3) These exceptions arise mostly due to coding mistakes in our program.	
4) ArrayIndexOutOfBoundsException, ClassCastException, IndexOutOfBoundsException	SQLException, FileNotFoundException, ClassNotFoundException

69) What is default Exception handling in java?

When JVM detects exception causing code, it constructs a new exception handling object by including the following information.

- 1) Name of Exception
- 2) Description about the Exception
- 3) Location of Exception.

After creation of object by JVM it checks whether there is exception handling code or not. If there is exception handling code then exception handles and continues the program. If there is no exception handling code JVM give the responsibility of exception handling to default handler and terminates abruptly.

Default Exception handler displays description of exception, prints the stacktrace and location of exception and terminates the program.

Note : The main disadvantage of this default exception handling is program terminates abruptly.