|  |
| --- |
| 1. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.  Object-oriented programming(OOPs) is a methodology that simplify software development and maintenance by providing some rules.  Basic concepts of OOPs are: |
| 1. Object 2. Class 3. Inheritance 4. Polymorphism 5. Abstraction 6. Encapsulation   2. Access Specifier:-The access specifier determines how accessible the field is to code in other classes. Access ranges from totally accessible to totally inaccessible.  Access Specifier are of 4 types in java   Public:-Anything declared as public can be accessed from anywhere means outside the class accessibility will be possible   Private:-Anything declared as private can not be accessed outside the class.   Protected:- Anything declared as protected can be accessed inside the same package and subclasses of the different package   default:-If you not declared anything in front of any variable then it will be declared as default automatically. Anything declared as default can be accessed only to classes in the same package.   3. The **static keyword** in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than instance of the class.  The static can be:   1. variable (also known as class variable) 2. method (also known as class method) 3. block 4. nested class   4.Array is a collection of similar type of elements that have contiguous memory location.It is an object that contains elements of similar data type.  It is a data structure where we store similar elements.  We can store only fixed set of elements in an array. Array is index based,first element of array is stored at 0 index.   1. Variable is a name of reserved area allocated in memory. It is a name of memory location. Variable means its value can be changed. Eg: int data=50;//data is variable. |

1. Data types represent the different values to be stored in the variable. It specifies the size and type of values that can be stored in an identifier.

There are 8 primitive data types:

* Boolean
* Char
* Byte
* Short
* int
* Long
* Float
* Double

The non primitive data types are Array,String,etc.

1. Operator is a symbol and it required in programming to perform operations on data.

The types of operators are:

**Unary Operator**

Postfix- exp++

Exp--

Prefix- ++expr

--expr

+expr

-expr

~expr

!expr

**Artihmetic Operator**

Multiplicative-

\* / %

Additive-

+ -

**Shift operator-**

<<

>>

>>>

**Relational operator-**

**Comparison –**

<

>

<=

>=

Instanceof

**Equality-**

==

!=

**Bitwise operator-**

Bitwise AND - &

Bitwise exclusive OR - ^

Bitwise inclusive OR - |

**Logical operator-**

Logical AND-&&

Logical OR-||

**Ternary operator** -?:

**Assignment operator-** +=,-=,\*=,/=,%=&=,^=,|=,<<=,>>=,>>>=,=

1. Depending upon the situation we have to apply conditional logics inside the program. May be 1 condition, 2 condition or as many as conditions in the program.  
   Different ways to implement conditions in the program are:
   1. If
   2. If---else
   3. If---elseif---elseif---else
   4. Switch---case-break---case-break---default
2. In else if ladder, the control goes through the every else if statement until it finds true value of the statement or it comes to the end of the else if ladder. In case of switch case, as per the value of the switch, the control jumps to the corresponding case.

* The switch case is more compact than lot of nested else if. So, switch is considered to be more readable.
* The use of break statement in switch is essential but there is no need of use of break in else if ladder.
* The variable data type that can be used in expression of switch is integer only where as in else if ladder accepts integer type as well as character.
* Another difference between switch case and else if ladder is that the switch statement is considered to be less flexible than the else if ladder, because it allows only testing of a single expression against a list of discrete values.
* Since the compiler is capable of optimizing the switch statement, they are generally considered to be more efficient. Each case in switch statement is independent of the previous one. In case of else if ladder, the code needs to be processed in the order determined by the programmer.
* Switch case statement work on the basis of equality operator whereas else if ladder works on the basis of true false( zero/non-zero) basis.

10.Looping Contructs: There are also situation where we need to repeat the same task or process again and again in the program. This can be achieved by implementing loop.   
Different ways to implement loop in the program are.  
1. While

2. do-while

3. for

4.advanced for loop

11. Phases: In every loop there are 3 steps which helps to define the loop properly.

1. Starting of the loop or also called as initialization phase.

2. Checking the end point of loop or how long the loop will work is called as testing phase.

3. Taking the loop to next level by incrementing or decrementing the counter or taking valid input from user to take loop at next repeatations.

12.

|  |  |  |
| --- | --- | --- |
| For | while | Do-while |
| Syntax:  For(initialization;condition;updating){  Statements;} | Syntax:  While(condition){  Statements;} | Do{  Statements;}  While(condition); |
| It is known as entry controlled loop | It is known as entry controlled loop | It is known as exit controlled loop |
| If the condition is not true for the first time then the control will never enter in a loop | If the condition is not true for the first time then the control will never enter in a loop | Even if the condition is not true for the first time, the control will enter in a loop |
| There is no semicolon after the condition in the syntax of the for loop | There is no semicolon after the condition in the syntax of the while loop | There is semicolon after the condition in the syntax of the do while loop |
| Initialization and updating is the part of the syntax. | Initialization and updating is not the part of the syntax. | Initialization and updating is not the part of the syntax. |

13. Array is a set of continuous memory location which is used to store the similar set of data inside it.

1. All the memory allocated to store the data is in continuous manner.

2. one logical name of variable is used to refer the memory location along with element number or subscript value.

3. we can declare single dimension array or multi dimension array.

14. In c,c++ array is of value type.With the help of pointer variable we can take reference of array. Even name of array itself indicates it first location.

In java, Array works like object.

Array is of reference type.

15. The function return type defines the constraints the data type of the value returned from a subroutine or a method. The return type must be explicitly specified when declaring a function.

16. We can call a function using call-by-value and call-by-reference. These functions can be done by passing the parameters inside the function. By passing these parameters we can access the function by giving value on run time. Sometimes we pass objects as the parameters and it is known as call-by-reference. If we give value of primitive data-type then it is known as call-by-value.

17. The data type void actually refers to an object that does not have a value of any type. functions which only print a message and have no value to return. Such a function is used for its side effect and not for its value. In the function declaration and definition, we have indicated that the function does not return a value by using the data type void to show an empty type, i.e. no value. Similarly, when a function has no formal parameters, the keyword void is used in the function prototype and header to signify that there is no information passed to the function.

18. **String [] aa** is a **java** array of **String** objects. This means that the main function expects an array of **Strings**. This array of **strings** typically holds all command line **parameter arguments** passed in when the program is run from the command line.

19.

20. Exception handling is the alternate way of handling the program. It avoids stopping the program during run time error.

21. Exception handling is required to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why we use exception handling.

22. to handle the exceptions in java language, we use the below keywords:

Try

Catch

Throw

Throws

Finally

**Try:** it contains the main logic which may result into exception during the execution of the program.

**Catch:** it is following the try block and providing the reference of exceptions which may be thrown by try. One try can be followed by many catch. It takes reference exception class as parameter and also helps in customizing the exception messages.

**Finally:** it is also following the try block which may throw exception and helping in releasing the memory occupied by the objects used in try block which may throw exception. It always execute irrespective of exception.

**Throw:** It is used for user defined exception. It is used to explicitly throw an exception.

Throws: it is used to declare an exception. It gives an information to the programmer there may occur an exception so it is better for the programmer to provide the exception handling code so the normal flow is maintained.

23. The constructor is a block of code which is called when an instance of object is created and memory is allocated for the object.

24. It is a special type of method which is used to initialize the object.

It constructs the value at the time of object creation.

25. A constructor which has a specific number of parameters is called parameterized constructor. It is used to provide different values to the distinct objects.

26. 1. Constructor name must be same as its class name.

2. constructor must have no explicit return type.

27. By using super() function we can use the base class constructor in child class. As per the rule base class constructor is always getting executed first before the child class constructor. If we do not call any base class constructor in child class by default, default constructor will get executed before executing child class constructor.

The base class constructor is the super keyword in java.

1. super can be used to refer immediate parent class instance variable.
2. Super can be used to invoke immediate parent class method.
3. Super() can be used to invoke immediate parent class constructor.

28. What will be the flow of constructor execution while implementing inheritance?

Always base class first and then child class.

The flow of constructor execution while implementing inheritance is from base/parent class to derived/child class.   
When we create an object of a child class, then first base class constructor and then derived class constructor gets called implicitly.  
(We cannot call a class constructor explicitly using an object or manually).

29. Function overriding is when same function name from a base class can be used in derived class.

When we use the base class function inside child class to implement same functionality with different values or body content is called function overriding.

Public class shape {

Public void draw() {

//somefunction

}

}

Public class circle extends shape {

Public void draw() {

//some different function than class shape

}

}

Method draw() in class circle overrides implementation of class shape.This function is called overriding.

30.Function overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different. Function overloading is required to increase the readability of the program.

It can be implemented by,

* By changing the number of arguments
* By changing the data type

Suppose if you have to perform addition of the given numbers but there can be any number of arguments, if you write the methods such as a(int,int) for two parameters and b(int,int,int) for three parameters then it may be difficult to understand the behavior of the method because its name differs. Hence, we use function overloading.

31. Inheritance is a mechanism in which one object acquires all the properties and behaviors of parent object.  
You can create new classes that are built upon existing class. When you inherit from existing class you can reuse methods and fields of parent class, and you can add new methods and fields also.  
Inheritance represents IS-A relationship , also known as parent-child relationship.  
The extends keyword indicates that you are making a new class that derives from an existing class. Extends is to “increase the functionality”.   
Inhertiance is used for,

Method overriding

Code reusability

Types:

Single inheritance

Multilevel

Hierarchial

32. Class can be inherited in java using extends.

Interface can be implemented in java using implements.

33. Interface in java is like abstract class which contain only abstract methods. Interface is used to define the functionality based upon various concepts.

|  |  |
| --- | --- |
| Abstract class | Interface |
| Abstract class can have abstract and non-abstract methods. | Interface can have only abstract methods. |
| Abstract class does not support multiple inheritance. | Interface supports multiple inheritance. |
| Abstract class can have final,non-final,static and non-static variables. | Interfaces has only static and final variables. |
| It can provide the implementation of interface. | It cannot provide the implementation of abstract class. |
| The abstract keyword is used to declare abstract class. | Interface keyword is used to declare interface. |
| It cannot be instantiated | It can be instantiated with help of reference. |
| Eg:  public abstract class Shape{  public abstract void draw();} | Eg:  public interface  Drawable{  Void draw();  } |

34. Frameworks are usually many classes of prewritten code to which you add your own code to solve a problem in specific domain. Framework is used by calling its methods, inheritance and supplying listeners or other implementations.

It provides readymade architecture for any types of programming which contains various sets of classes and interfaces to provide programming support and automating tasks. It contains various API’s and programming architectures which helps programmers to introduce standard modules inside softwares.

35. Collection in java is a framework that provides an architecture to store and manipulate the group of objects.

Java collections are like containers which help to group multiple items in single unit.

Java supports the implementation of Data Structures concepts through java collections. Unlike C and C++ programming language where we have to create the data structure objects before using it inside the application, java supports readymade objects which is helping programmers to manage the data in their application efficiently.

Whatever data is stored inside java collection object can be accessed and manipulated with the help of readymade methods supported by java collections. Main use of java collections is to store and process a set of data like data structure in c and c++ languages.

Java supports various methods,classes and functions to store and manage the data properly in the form of java objects.

36. Collection framework provides a standard architecture for storing and manipulating group of objects It provides various classes,interfaces and methods for these tasks. It also supports various algorithms used in achieving this objective. It is supported by java.util package

37. It uses dynamic array for storing the elements. It inherits abstract list class and implements list interface.

38. Iterator is an interface which provides iterator() to iterate on collection. It read the entire collection classes  
It works in single directional and it is universal cursor.It read and remove the object value.  
It has two methods:

* hasNext()-returns true if next element is present else false.
* next()-return the next element

39. Array list contains duplicate elements. It maintains insertion order and it is non-synchronized. It allows random access because array works at the index basis. Manipulation is slow because a lot of shifting occurs if any element is removed from the array list.

40. In array list the size is optional. We can use generics for type safety.

41.

42.It uses doubly linked list to store the elements . It inherits the abstractlist class and implements list and deque interfaces.

43. ListIterator is used to traverse the element in backward and forward direction. It can read,remove and replace object value . It has next(),hasNext(),previous(),hasPrevious() methods.

44. Linked list contains duplicate values.

Linked list class maintains insertion order.

It is non-synchronized.

Manipulation is fast because no shifting needs to be occurred.

It is used in list,stack,queue

45. Manipulations is fast and It can be used as list, stack or queue. It uses doubly linked list to store the elements. We can add or remove elements from both the sides.

46. The history section of web browsers where it creates a linked list of web-pages visited so that when you check history (traversal of a list) or press back button, the previous nodes data is fetched.

47. Arraylist is based upon the concept of dynamic array where as linked list is doubly linked list.

Arraylist is slow in handling data whereas linked list is fast.

Arraylist can act as list whereas linked list can act as list and queue.

Arraylist is better for storing and accessing the data whereas linked list is better for manipulating the data.

48. Generic is used to store the specific type of objects . A class that refer to any type is known as generic class.

49. It provides

type safety – we can store only one type of objects

type casting is not required

compile type checking

50.

51. HashSet is used to create a collection that uses a hash table for storage. It inherits the AbstractSet class and implements the Set interface.

52.

|  |  |
| --- | --- |
| List | Set |
| List is an ordered collection. It maintains insertion order | Set is an unordered collection . It doesn’t maintain any order. There are few implementation of sets which maintains order such as Linked HashSet |
| List allows duplicates | Set doesn’t allow duplicates.It would replace the existing duplicate values if inserted. |
| List Implementations: ArrayList, LinkedList | Set Implementations: Hash Set,Linked HashSet,Treeset,etc |
| List allows any number of null values. | Set can have only a single null value at most |
| List Iterator can be used to traverse a List in both the directions. | Iterator is used to traverse a set in only one direction. |
| List interface has one legacy class called vector | Set interface doesn’t have any legacy class |

53. Hashset stores the elements by using mechanism called hashing.

Hashset contains unique elements only.

54. Hash set contains only unique elements. If a duplicate element is inserted it replace it.

55.

56. The purpose of constructor is to initialize the object of the class. It is used for efficient memory management. It constructs a value at the time of object creation.

1. Constructors are not methods and they don’t have any return type. Constructor name should match with class name. Constructor can use any access specifier. Constructor overriding is not possible. Interfaces don’t have constructors

58.

|  |  |
| --- | --- |
| Constructors | Functions |
| Constructors create and initialize objects that doesn’t exists yet | Functions perform operations on objects that already exists. |
| Constructors cant be called directly; they are called implicitly when the new keyword creates an object. | Functions can be called directly on an object that has already been created with new. |
| Constructors must be named with the same name as the class name. | Functions need not be same as the class name. |
| They can’t return anything ,even void. | Functions must be declared to return something, although it can be void |

59. Inheritance helps programmer to implement the concept of generalization.

General properties are the part of base class and special properties are the part of child class. When child class inherits base class general properties are automatically getting added inside the child class without writing the code.

Inheritance is a mechanism in which one object acquires all the properties and behaviors of parent object.

60. Inheritance is implemented by using the keyword ‘extends’ and super keyword.

Class subclass-name extends Superclass name

{

//methods and fields

}

61. Access specifier governs the access of the variables and methods by other subclasses or methods. Private method or variable cannot be accessed by the subclass. Other access specifier have no significance as they can be accessed by the sub class.

62. To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

Consider a scenario of three classes A,B, and C. C inherits A and B. If A and B class have same method and you call it from child class object, there will be ambiguity to call method of classes A or B.

In order to overcome this errors multiple inheritance is not supported in java.

Hence, we use interfaces.

63. this keyword represents the reference of class within the class itself or within the class scope.

Super keyword represents the reference of the class outside the base class, within child class. It can also be used as base class constructor.

64. when one task is performed by different ways that is known as polymorphism.

Benefits: It provides flexibility to the programmer to write programs that uses single method for different operations.

65. Polymorphism is achieved by 1.Method overloading and

2. Method overriding

66. Method overloading is used to increase the readability of the program and it is performed within a class. It must have different parameters. Method overloading is an example of compile time polymorphism. Method overloading cannot be performed by changing the return type of the method only. Return type can be same or different, but you must have to change the parameter.

It is a process of creating more than one method in same class with same name but different parameter.

Rules:

1. Number of arguments in a method should be different.
2. Type of argument in a method should be different(if number of argument in a method are same)
3. Function (method) name should be same.
4. Return type can be nothing.

67.Method overriding is used to provide the specific implementation of the method that is already provided by its super class. It occurs in two classes that have IS-A(inheritance)relationship. It must have same parameters. It is an example of run time polymorphism. Return type must be same or covariant in method overriding.

It is the process of creating multiple method with same name and same parameter in base as well as child class with different body.

Rules:

1. The method signature in overridden and overriding methods must be exactly same. If they are different, then it will be an overloaded method.
2. The return type of overriding method can be child class of return type declared in overridden method.
3. Private,static,final methods cannot be overridden in java.
4. Overriding method cannot throw checked exception higher in hierarchy than thrown by overridden method.
5. Overriding method cannot reduce the access scope of overridden method. If overridden method in parent class is protected, then overriding method in child class cannot be private. It must be either protected (same access) or public.
6. Must be a is-a relationship.
7. Must have same parameters as in the parent class.

68. The final keyword in java is used to restrict the user. It is used to define an entity that can be assigned only once. Once a final variable has been assigned, it always contains the same value.

It stops value change, stops overriding and stops inheritance.