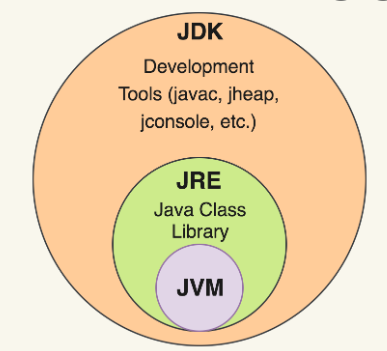
**1. Explain JDK, JRE and JVM?**

|  |  |  |
| --- | --- | --- |
| **JDK** | **JRE** | **JVM** |
| JDK stands for Java Development Kit | JRE stands for Java Runtime Environment | JVM Stands for Java Virtual Machine |
| It contains JRE + development tools (javac, javap etc.). | It contains JVM + Java class library. | It provides environment to run byte code. |



**2.Explain public static void main(String args[]) in Java?**

public:

Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible from anywhere. JVM calls main methos hence it must be static.

Static

We can call static resource using class name. JVM calls main method using class name hence it must be static.

void

It is the return type of the method. void defines the method which will not return any value.

main

It is the name of the method which is searched by JVM.

String args[]

It is a string array used to store command line arguments.

**3.Why Java is platform independent?**

Java is called platform independent because of its byte codes which can run on any operating system.

**4.Why Java is not pure object oriented language?**

Java is not pure object oriented language because it makes use of eight primitive data types such as byte, short, int, long, float, double, char, boolean which are not objects.

**5.What are wrapper classes in Java?**

Wrapper classes convert the Java primitive types into the reference types.

|  |  |
| --- | --- |
| **Primitive Types** | **Wrapper Class** |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| float | Float |
| double | Double |
| boolean | Boolean |
| char | Character |

**6. What is Constructor?**

* Constructor name and class name must be same.
* Constructor doesn’t have return type.
* When object is created constructor gets invoked automatically.
* Constructor is used to initialize object.

**7. What is the difference between Method and Constructor?**

|  |  |
| --- | --- |
| **Constructor** | **Method** |
| Constructor name and class name must be same. | Method name and class name may or may not be same. |
| Constructor doesn’t have return type. | Return type is compulsory. |
| Constructor execution is implicit process. | Method execution is explicit process. |
| Constructor is used to initialize object. | Method is used write business logic. |

**8. How many types of Constructors are in Java?**

1. 0-arg constructor
   * Constructor without parameter is called 0-arg constructor.
   * If we want to assign same value to all object.
2. Parameterized Constructor
   * Constructor with parameter is called parameterized constructor.
   * If we want to assign different value to each object.

**9. What is a private constructor?**

* To create a constructor as private, use the private keyword while declaring it. It can only be accessed within that class.
* We need a private constructor while creating Singleton Class.

**10. What is Singleton class?**

* Java class for which we can create only one object is called singleton class.

**11. Write a program to demonstrate singleton class.**

class Test{

private static Test t = null;

private Test() {

System.out.println("Constructor...");

}

static Test create() {

if(t==null) {

t=new Test();

}

return t;

}

}

public class Main {

public static void main(String[] args) {

Test t1 = Test.create();

Test t2 = Test.create();

}

}

**12. Can we make a constructor final?**

* No, we cannot make a constructor final. If we made a constructor final, it would throw a compile-time error "modifier final not allowed".

**13. What is the use final modifier?**

* Final modifier always provides restriction.
* Final modifier can be applied with variable, method and class.
* Final variable: Not able to change value.
* Final method: Prevents method overriding.
* Final: Class: Prevents Inheritance.

**14. Can we make a constructor static?**

* The constructors cannot be static.

**15. What are the various access modifiers for Java classes?**

* Private
* Default
* Protected
* Public

**16. What is the difference between an Inner Class and a Sub-Class?**

* An Inner class is a class which is present within another class.
* Sub-class is a class which inherits from another class.

**17. What is an infinite Loop? How infinite loop is declared?**

* It executes infinitely.

Approach1:

**for**(;;) {

System.***out***.println("welcome...");

}

Approach2:

**for**(;1==1;) {

System.***out***.println("welcome...");

}

Approach3:

**while**(**true**) {

System.***out***.println("welcome...");

}

**18.What is the difference between continue and break statement?**

* Break is used to terminate the loop.
* Continue is used to skip the current iteration of the loop.

**19. Give an example of ternary operator.**

class Main{

public static void main(string args[]){

int a=10,b=10,result;

result=a>b?a:b;

System.out.println("Greater No: "+result);

}

}

**20. How can you generate OTP (6 digit) in Java?**

import java.util.Random;

public class OTP {

public static void main(String[] args) {

Random r = new Random();

String code="";

for(int i=1;i<=6;i++){

code = code + r.nextInt(9);

}

System.out.println(code);

}

}

**21. What’s the difference between an Abstract Class and Interface in Java?**

* Abstract class contains both defined method and undefined method.
* Interface contains only undefined class.

**22. Does Importing a package imports its sub-packages as well in Java?**

* No

**23. When can you use the super keyword?**

* Super represents reference of an immediate parent class.
* The super can be used to call the method of an immediate parent class.

**24. When can you use the keyword?**

* this represents reference of current class.
* this can be used to call the method, variable of current class.

**25. How is the creation of a String using new operator different from that of without new operator?**

* Without new operator object will be created in SCP (String Constant Pool) area.
* With new operator object is created in heap area.

**26. What is a package in Java? List down various advantages of packages.**

* Package is a collection of class and interfaces.
* Advantages:
  + Packages help in avoiding name conflict
  + Easy to access classes and interfaces

**27. What is inheritance in Java? Explain different types of inheritance**

* It is the process of getting property (variable) and behavior (method) from one class to another class.
* Types of Inheritance:
  + Single
  + Multilevel
  + Hierarchical
  + Multiple (not supported in case of class)
  + Hybrid

**28. What is difference between method overloading and method overriding?**

|  |  |
| --- | --- |
| **Method Overloading** | **Method Overriding** |
| Class having more than one method with same name but different parameters. | Redefining parent class method in child class is method overriding. |

**28. What if I write static public void instead of public static void?**

* The program compiles and runs correctly because the order of modifiers doesn't matter.

**29. What is the default value of the local variables?**

* The local variables are not initialized to any default value.

**30. What is the output of below program?**

class Test

{

public static void main (String args[])

{

System.out.println(10 + 20 + "Java");

System.out.println("Java" + 10 + 20);

}

}

Output: 30Java, Java1020

**31. What is the output of below program?**

class Test

{

public static void main (String args[])

{

System.out.println(10 \* 20 + "Java");

System.out.println("Java" + 10 \* 20);

}

}

Output:

**32. Write a Java Program to reverse a string without using String inbuilt functions.**

import java.util.Scanner;

public class ReverseString {

public static void main(String[] args) {

String rev=" ";

Scanner sc= new Scanner(System.in);

System.out.print("enter the string:");

String str=sc.nextLine();

int l=str.length();

for(int i=l-1;i>=0;i--) {

rev=rev+str.charAt(i);

}

System.out.println("Reverse String:"+rev);

}

}

**33. Does an empty file name with .java extension a valid file name?**

* Yes, Java permits to save our java file by .java only. It is compiled by javac .java and run by java class name.

**34. What do you mean by Collections in Java?**

* It represent group of object in a single entity.

**35. Differentiate between == and equals() ?**

* == Operator used for reference comparison.
* Equals () is used for content comparison.

**36. Which class is the super class for all the classes?**

* The object class is the superclass of all other classes in Java.

**37. Why is multiple inheritance not supported in java in case of class?**

* Multiple inheritance not supported because it creates ambiguity problems.

**38. Can we use this() and super() both in a constructor?**

* No, because this() and super() must be the first statement in the class constructor.

**39. What is the output of the following Java program?**

class Base

{

public void baseMethod()

{

System.out.println("BaseMethod called ...");

}

}

class Derived extends Base

{

public void baseMethod()

{

System.out.println("Derived method called ...");

}

}

public class Test

{

public static void main (String args[])

{

Base b = new Derived();

b.baseMethod();

}

}

**Output:** Derived Method Called…

**40. What is the output of the following programs.**

class Base

{

protected final void getInfo()

{

System.out.println("method of Base class");

}

}

public class Derived extends Base

{

protected final void getInfo()

{

System.out.println("method of Derived class");

}

public static void main(String[] args)

{

Base obj = new Base();

obj.getInfo();

}

}

**Output:** Error

**41. Can we declare an interface as final?**

* No, we cannot declare an interface as final because the interface must be implemented by some class to provide its definition.

**What is the meaning of immutable regarding String?**

* The meaning of immutable is unmodifiable or unchangeable. String is immutable, i.e., once string object has been created, its value can't be changed.

**How many objects will be created in the following code?**

String s1="BBSR";

String s2="BBSR";

String s3="CDAC";

**Answer:** 2 Objects

**What is the output of the following Java program?**

public class Test {

public static void main (String args[])

{

String a = new String("BBSR");

String b = "BBSR";

if(a == b)

{

System.out.println("a == b");

}

if(a.equals(b))

{

System.out.println("a equals b");

}

}

}

**Output:** a equals b

**. What are the differences between String and StringBuffer?**

|  |  |
| --- | --- |
| **String** | **StringBuffer** |
| String is immutable. | StringBuffer is mutable. |
| We can creates string with new and without new operator. | StringBuffer is created by new operator only |
| String class equals() is used for content comparison. | StringBuffer class equals() is used for reference comparison |

**What are the differences between StringBuffer and StringBuilder?**

|  |  |
| --- | --- |
| **StringBuffer** | **StringBuilder** |
| StringBuffer method’s are synchronized | StringBuilder methods are not synchronized |
| Thread Safe | Thread Unsafe |
| StringBuffer is less efficient than StringBuilder. | StringBuilder is more efficient than StringBuffer. |

**Write a Java program to count the number of words present in a string?**

public class WordDemo { public static void main(String[] args) { String s = "My Life My Rules"; String arr[]= s.split(" "); System.out.println("Total word in String is :"+arr.length); }}

**Write a Java program to count the number of characters present in a string?**

public class Test {

static public void main(String[] args) {

String name = "BSSR";

System.out.println("No. Char: " + name.length());

}

}

**What are shallow copy and deep copy in java?**

* Shallow Copy
  + The shallow copy only creates a new reference and points to the same object.
* Deep Copy
  + In a deep copy, we create a new object and copy the old object value to the new object.

Example: Shallow Copy

**class** Student{

String name ="Raj";

**int** age = 30;

}

**public** **class** ShallowCopy {

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

Student s1 = **new** Student();

Student s2 = s1;

System.***out***.println(s1.name+"::::"+s1.age);

System.***out***.println(s2.name+"::::"+s2.age);

}

}

Example: Deep Copy

**class** Student{

String name;

**int** age;

}

**public** **class** DeepCopy {

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

Student s1 = **new** Student();

s1.name = "Ram";

s1.age = 30;

Student s2 = **new** Student();

s2.name = s1.name;

s2.age = s1.age;

System.***out***.println(s1.name+"::::"+s1.age);

System.***out***.println(s2.name+"::::"+s2.age);

}

}

**What is the main objective of garbage collection?**

* The main objective of this process is to free up the memory space by deleting unused objects.

**Who is responsible to perform Garbage collection?**

* Garbage collector is responsible for performing garbage collection.

**Is it possible to call garbage collector by the programmer? Explain in brief.**

* No, Programmer cannot call garbage collector.
* JVM calls garbage collector in random interval.
* Programmer can request JVM to call garbage collector.

**. How programmer can request JVM to call Garbage Collector?**

* There are 2 different ways:
  + Using System.gc()
  + Using Runtime.getRuntime().gc()

**. How to make object eligible for Garbage Collection?**

* Nullifying Reference Variable
* Reassign Reference Variable
* Island of isolation

**What part of memory - Stack or Heap - is cleaned in garbage collection process?**

* Heap Memory

**. Is it possible that the ‘finally’ block will not be executed?**

Yes, if we shutdown JVM then it will not execute.

**public** **class** Main {

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

**try** {

//shutdown JVM

System.*exit*(0);

}

**catch**(Exception e) {

System.***out***.println("catch.....");

}

**finally** {

System.***out***.println("finally.....");

}

}

}

**What is the difference between Object-oriented and object-based programming language?**

* Object-oriented programming languages support all the features of OOPS including polymorphism and inheritance.
* Object-based programming languages support all the features of OOPS excluding polymorphism and inheritance.

**. What is the difference between inheritance and composition?**

* Inheritance
  + Inheritance is also knows as IS-A relationship.
  + We can establish inheritance relation using extends keyword.
* Composition
  + Composition is also known as HAS-A relationship
  + By creating referenced class object.

**. Can we import the same package/class twice? Will the JVM load the package twice at runtime?**

* Yes, we can import same package twice but JVM loads only once into the memory.

**. Distinguish between static loading and dynamic class loading.**

Static Loading

Test ob = new Test();

Dynamic Loading

Class.forName(“Test”).newInstance()

**How can we find the actual size of an object on the heap?**

* In Java, there is no way to find out the actual size of an object on the heap.

**. What are the functions of hashCode() method?**

* The hashCode() method returns address of an object.

**What is Polymorphism?**

* Single interface with many forms is called polymorphism.
* Example:
  + Method Overloading (Compile time Polymorphism)
  + Method Overriding (Runtime Polymorphism)

**. Explain data encapsulation in Java?**

* Encapsulation wraps the data (variables) and code(methods) as a single unit.

**class** Test{

**private** **int** age;

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **int** getAge() {

**return** age;

}

}

**public** **class** Main {

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

Test t = **new** Test();

t.setAge(10);

System.***out***.println(t.getAge());

}

}

**. Explain the thread lifecycle in Java?**

* New
* Ready
* Running
* Blocked
* Dead

**What is the difference between Array list and vector in Java?**

|  |  |
| --- | --- |
| **ArrayList** | **Vector** |
| Array list is not synchronized | Vector is synchronized |
| Performance is better | Less performance compared to ArrayList |
| Thread Unsafe | Thread Safe |
| New Object Size:  old capacity = new capacity\*3/2+1 | New Object Size:  old capacity = 2\* newcapacity |

**. What is Serialization?**

* The process of writing state of an object into a file is called Serialization.

**What is De-Serialization?**

* The process of reading state of an object from a file is called De-Serialization.
* **What is the difference between Array and ArrayList in Java?**

|  |  |
| --- | --- |
| **Array** | **ArrayList** |
| Array contains only homogenous data elements. | ArrayList contains both homogenous and heterogeneous data elements. |
| Array is static in size. | ArrayList is dynamic in size. |
| Arrays can contain primitive data types as well as objects | ArrayList can contain only object type. |

**What is a marker interface?**

* Empty interface is called marker interface.
* Example:
  + Serializable

**. What is Functional Interface?**

* Interface in which having only one abstract method is called functional interface.
* Example:
  + Runnable

**EXEPTION HANDLING**

**1. What is an exception?**

* Error occurred at runtime is called Runtime error. Runtime error is also known as exception.

**2. What is Exception handling in Java?**

* Exception Handling is a mechanism to handle run time errors.
* It is used to avoid abnormal program termination.

**3. How the exceptions are handled in Java?**

* Exceptions are handled using try, catch and finally blocks.

**4. What is the difference between error and exception in Java?**

* Errors are mainly caused by the environment in which an application is running.
* Exceptions are mainly caused by the application itself.

**5. Can we write only try block without catch and finally blocks?**

* No, try block required either catch block or finally block.

**6. What is the super or base class of all exceptions in Java?**

* Exception is the super class of all exceptions in Java.

**7. What is the super class for error and exception classes in Java?**

* Throwable is the super class for error and exception classes in Java.

**8. What are the types of exceptions?**

* Predefined Exception
* User Defined / Custom Exception

**9. Which exception may be thrown if the given code is executed?**

class Test

{

public static void main(String[] args)

{

int arr[] = new int[5];

System.out.println(arr[6]);

}

}

ArrayIndexOutOfBoundsException

**10. Which exception may be thrown if the given code is executed?**

class Test

{

public static void main(String[] args)

{

int a = 20;

int b = 30;

int c = 10;

int x = (a \* b)/(a - b + c);

System.out.println("Result: " +x);

}

}

ArithmeticException

**11. Explain the Java exception hierarchy.**

* Throwable class is the parent class of all exception types.
* Throwable class has two subclasses Error and Exception.

**12. What does JVM do when an exception occurs in a program?**

* When JVM gets an exception in a program, it creates an exception object and throws it.
* If the exception object is not caught properly then JVM terminates program abnormally.

**13. What is the difference between checked and unchecked exceptions?**

* Exception which is checked by compiler is called checked exception.
* Exception which is not checked by compiler is called unchecked exception.

**14. How will you handle the checked exception?**

* A checked exception can be handled either by using try and catch block or by using throws.

**15. What is unreachable catch block error?**

class ExceptionHandling

{

public static void main(String[] args)

{

try

{

}

catch(Exception ex)

{

}

catch(NumberFormatException ex)

{

//Compile time error

//This block becomes unreachable as

//exception is already caught by above catch block

}

}

}

**16. What is the use of throws keyword in Java?**

* It used to bypass the exception.

**17. What is the use of throw keyword in Java?**

* throw is used to throw exception (predefined or user defined) explicitly.

**18. What is the difference between final, finally and finalize?**

* final:
  + It is a keyword.
  + Final keyword provides restriction.
* finally:
  + It is a block.
  + It is given to perform clean up operation.
* finalize:
  + It is a method.
  + Before destroying object finalize method call once to perform object cleaning.

**19. What is StackOverflowError in Java?**

* When stack memory is full and we try to push element in stack that time will get stack over flow error.

**20. What is OutOfMemoryError in Java?**

* Unable to allocate memory to an object due to insufficient space in the heap area.

**21. Give some examples to checked exceptions?**

* IOException, FileNotFoundException, ClassNotFoundException etc.

**22. Give some examples to unchecked exceptions?**

* ArrayIndexOutOfBoundsException, NumberFormatException, ArithmeticException, IllegalArgumentException etc.

**23. What is the use of printStackTrace() method?**

* printStackTrace() method is used to print the detailed information about the exception occurred.

System.out.println(e.printStackTrace());

**24. How do we create customized exceptions?**

* Create class by extending Exception class.
* Take parameterized constructor.
* Call parent class constructor using super

**25. What are the legal combinations of try, catch and finally blocks?**

* try – catch
* try – finally
* try – catch – finally