1.who is develop java programming

Ans. java programming developed by sun microsystem in 1990 for used and java used to primarily used for internet based application.
Release in 1995

2.why is java so papular
Ans. Java is easy to program and easy to run and platform independent which means it can be used mobile application and desktop application and that run different and server and application.and mainly reason write once and run anywhere.

3.what is platform independent
Ans. java platform independent means java compile code (byte code)can run all operating system so this is the reason java is platform independent.

4.what is byte code in java.
Ans.byte code as a define as intermediate code generated by compiler after compilation of java program (source code) intermediate code makes java platform independent.

5.compare jdk and jvm and jre?
ans.
Jdk. jdk means java development kit this kit provide the environment develop and execute the program.
Jdk is kit (package) that two things.
Development toos (to provides the environment develop your java program)
jre(to execute your java program)

jRE. Jre means java runs time enviorment is installation package that provide the enviorment to only run (not develop) java program on your computer jre is used to who want to run java program they are the end users of your system .

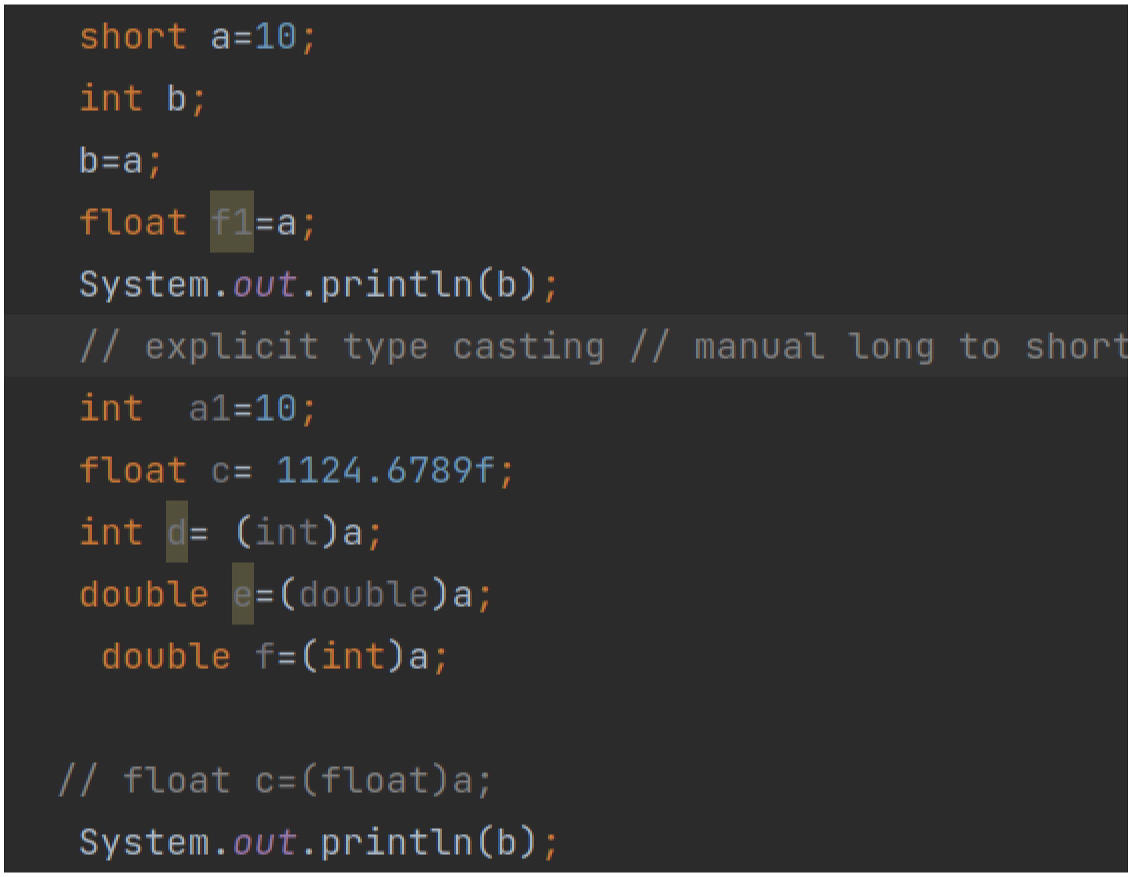
Jvm . java virtual machine this is very important both of because java jvm contain both whatever java program you run jre and jdk .program goes to jvm that is responsible for all java program and line by lin e and this is also known as interpreter

6.what is taken ?how many type \

ans.Taken is the smallest unit and individual building block in java.java compiler uses it .
There are 6 type of token
Keyword identifier,literal operator ,seperator ,white space.
7. What is datatype and its type
Ans data type are two type primitive and non primitive primitive data type are 8 type int char date type means we are store the value.

8.what are the different type of operator
ans.Airthmatic ,ralational ,logical,assignment ,bitwise ,uniray,shifft,instaceof,ternary.

9.What is casting
Ans.casting means you assign the value one type to another type their is type o casting



```
   short a=10;
   int b;
   b=a;
   float f1=a;
   System.out.println(b);
   // explicit type casting // manual long to short
   int  a1=10;
   float c= 1124.6789f;
   int d= (int)a;
   double e=(double)a;
    double f=(int)a;

  // float c=(float)a;
   System.out.println(b);
```

1.windiring casting(automatic casting)2.convert smaller size to larger size byte>shor>char>int >long>float>double
What is implicit type casting in java.
Ans.implicit type conversion means does’t lose its original position.

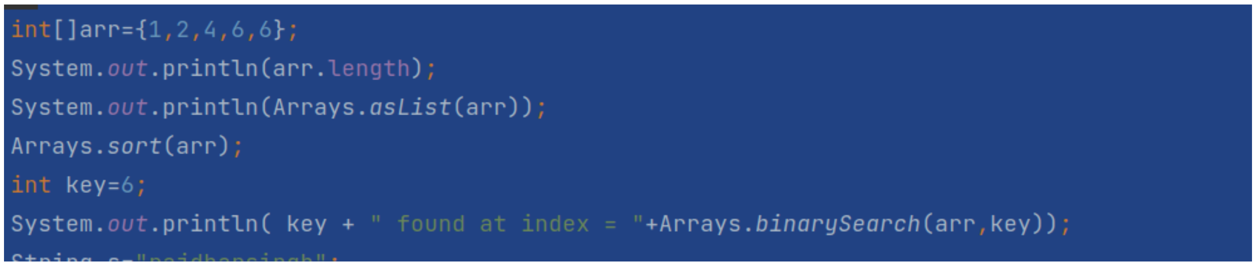
11. What is the expillit type casting
Ans.force type conversion is called explicit type casting ex. `(type-name) expression`

```
   float a = 1.2;
   //int b  = a; //Compiler will throw
an error for this
   int b = (int)a + 1;
```

12.what is array and different type array\

Ans.array is the collection of similar type of array the size is fixed for the array size is never changed this is called array there are two type of array
Single dimension and multidimensional array
Int []arr ,int [][]arr

13.what is array method in java
Ans.Array class in java that is useful method in common array like sorting searching updation in java
The will all method perform on array static nature not
Need to create instance of arry class



```
int[]arr={1,2,4,6,6};
System.out.println(arr.length);
System.out.println(Arrays.asList(arr));
Arrays.sort(arr);
int key=6;
System.out.println( key + " found at index = "+Arrays.binarySearch(arr,key));
```

.

14. What is jagged array

Ans.jagged array is the special type of multidimensional array which have variable number of columns.it is an array of array where each element in array can be different size.

```
int[][] Jagged_arr = {
    { 99, 18, 1, 77 },
    { 43, 8 },
    { 17,101,2 } };
```

15. What is difference between length and length()
Ans. difference between length tells the array of length length() method tells how many character contains is in string.

16.what is class
Ans. class is group of object that share the common property of and attribute
Class is not a real word entity class dose not take any space.
17. What is object
Ans.object is the blueprint of object i is a real word entity and they will take memory .
18.what is behavior of object
Ans.behavior means method are fuction of object

19.what is state of object

Ans.object means property of variable is called state

20. What is the different type of variable in java.
Ans There are three different type of variable in java
Local instance and static
Local . this variable define the inside of the body of method
instance.This method are declare outside the method of body this variabe define without using the static keyword.

Static variable.This variable used only once they program execution start . it is variable that should be initialize the first and before the instance variable.

21 . what is the different type of method in java
Ans. there is the two type of method in java
Predefined and userdefined method in java
Predefined method means they are build in method in java library is called predefined
Userdefined method means user write own method in java they will perform the own task.

22.what is the inheritance in java

Ans.inheritace in java child acquire all the property and method in with the help of extends keyword in this is called inheritance in java.
There are different type of inheritance in java single ,doubl ,multilevel ,hybrid hirichal

23.what is method overloading in java
Ans. method overloading means method names are same but parameter are different is called method overloading

24. What is method overriding in java
Ans.same name as method is called method overriding.

25. What is method hiding
Ans.the super class and subclass having same method name as parameter if they are static
The method superclass will be hidden method that is the subclass this mechanism is known as method hiding

26. Can you achieve method overriding on static method

Ans. no we can’t override the static method in java because static method is based on dynamic binding in runtime and static method is bounded in compile time so we can’t override static method in java.

27 what is upcasting
Ans.upcasting is the type of object casting of child object to perent class objects we can access the variable method parent class to the child class.

28 Difference between Upcasting and Downcasting

These are the following differences between Upcasting and Downcasting:

S.No	Upcasting	Downcasting
1.	A child object is typecasted to a parent object.	The reference of the parent class object is passed to the child class.
2.	We can perform Upcasting implicitly or explicitly.	Implicitly Downcasting is not possible.
3.	In the child class, we can access the methods and variables of the parent class.	The methods and variables of both the classes(parent and child) can be accessed.

4.	We can access some specified methods of the child class.	All the methods and variables of both classes can be accessed by performing downcasting.
5.	Parent p = new Parent()	Parent p = new Child() Child c = (Child)p;

---

28. Can super class reference variable hold an object of sub class

Ans. A reference variable of a superclass can be assigned a reference to any subclass derived from that superclass.

```java
class Data {

    int data1;
    int data2;
}

class NewData extends Data{

    int data3;
    int data4;
}

public class Javaapp {

    public static void main(String[] args) {

        Data obj = new NewData();
        obj.data1 = 50;
        obj.data2 = 100;
        System.out.println("obj.data1 = "+obj.data1);
        System.out.println("obj.data2 = "+obj.data2);
    }
}
```

29. Is multiple inheritance is allow in java

Ans.no multiple inheritace is not allow in java because they will create a diamond problem there is class c extends the a and b class there is same method a and b display() then java compiler is not understand which method is extends so ambiguity will generate this is not support multiple inheritance

30 what is constructor in java
Ans constructor is same as class name when we are create a object at least one constructor called it is called constructor

31. What is default constructor
Ans. default construtor is java we not need to create any construtor compiler create default construtor
Ex studnet s =new statudet();

```
public class Student {
  String firstName;
  String lastName;
  int age;

  public static void main(String
args[]) {
      Student myStudent = new
Student();

      myStudent.firstName =
"Ihechikara";
      myStudent.lastName = "Abba";
      myStudent.age = 100;


System.out.println(myStudent.age);
      //100



System.out.println(myStudent.firstName)
;
      //Ihechikara

  }
```

32 . how do you call a super class constructor from sub class constructor

Ans. To explicitly call the superclass constructor from the subclass constructor, we **use super()** . It's a special form of the super keyword. super() can be used only inside the subclass constructor and must be the first statement.

```
Class Person{
  public String name;
  public int age;
  public Person(String name, int age){
     this.name = name;
     this.age = age;
  }
  public void displayPerson() {
     System.out.println("Data of the Person class: ");
     System.out.println("Name: "+this.name);
     System.out.println("Age: "+this.age);
  }
}
public class Student extends Person {
  public String branch;
  public int Student_id;
  public Student(String name, int age, String branch, int Student_id){
     super(name, age);
     this.branch = branch;
     this.Student_id = Student_id;
  }
  public void displayStudent() {
     System.out.println("Data of the Student class: ");
     System.out.println("Name: "+this.name);
     System.out.println("Age: "+this.age);
     System.out.println("Branch: "+this.branch);
     System.out.println("Student ID: "+this.Student_id);
  }
  public static void main(String[] args) throws CloneNotSupportedException {
     Person person = new Student("Krishna", 20, "IT", 1256);
     person.displayPerson();
  }
}
```

33 What is the use of this keyword

ans.this keyword in Java

There can be a lot of usage of **Java this keyword**. In Java, this is a **reference variable** that refers to the current object.

Usage of Java this keyword

Here is given the 6 usage of java this keyword.

1. this can be used to refer current class instance variable.
2. this can be used to invoke current class method (implicitly)
3. this() can be used to invoke current class constructor.
4. this can be passed as an argument in the method call.
5. this can be passed as argument in the constructor call.
6. this can be used to return the current class instance from the method.

34.can a constructor be called directly from a method

Ans no you can't call the construtor dircelty from a method

```
public class Student {
  private String name;
  private int age;
  Student(){}
  Student(String name, int age){
    this.name = name;
    this.age = age;
  }
  public void SetValues(String name, int age){
    this(name, age);
  }
  public void display() {
    System.out.println("name: "+this.name);
    System.out.println("age: "+this.age);
  }
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the name of the student: ");
    String name = sc.nextLine();
    System.out.println("Enter the age of the student: ");
    int age = sc.nextInt();
    Student obj = new Student();
    obj.SetValues(name, age);
    obj.display();
  }
}
```

35.is a super class constructor callede even when there is no explicit call from a sub class constructor
Ans
36. What is the polymorphism in java
Ans. polymorphism means one task perform in different way it is called polymorphism
Runtime polymorphism and complitime polymorphism(method overriding)

36. What is use of instaceof operator in java
ans.The `instanceof` operator in Java is used to check whether an object is an instance of a particular class or not.

Java instanceof during Inheritance

We can use the `instanceof` operator to check if objects of the subclass is also an instance of the superclass.

The `instanceof` operator is also used to check whether an object of a class is also an instance of the interface implemented by the class

What is interface in java
Ans.interface is completely abstract class interface have abstract method thair no body of method we are used the interface method to use implements keyword this called interface.

39. What is encapsulation in java
Ans.data and code binding together in one sigle unit is called encapsulation

40.how do i define interface
Ans inference keyword to define interface

41. How do you implement interface
ans.The `implements` keyword is used to implement an `interface`.

The `interface` keyword is used to declare a special type of class that only contains abstract methods.

```
interface Animal {
  public void animalSound(); // interface method (does not have a
body)
  public void sleep(); // interface method (does not have a body)
}

// Pig "implements" the Animal interface
class Pig implements Animal {
  public void animalSound() {
    // The body of animalSound() is provided here
    System.out.println("The pig says: wee wee");
  }
  public void sleep() {
    // The body of sleep() is provided here
    System.out.println("Zzz");
  }
}

class MyMainClass {
  public static void main(String[] args) {
    Pig myPig = new Pig();  // Create a Pig object
    myPig.animalSound();
    myPig.sleep();
  }

}
```

44.can a class extend multiple interface
43.can u extend an interface
Ans.yes because two interface extends to each other
45.what is abstract class in java
Ans.abstract class restricted two create a object a class abstract class has a abstract method can only use in abstract class method does't have a body.

```
abstract class Bank{
abstract int getRateOfInterest();
}
class SBI extends Bank{
```

```
int getRateOfInterest(){return 7;}
}
class PNB extends Bank{
int getRateOfInterest(){return 8;}
}

class TestBank{
public static void main(String args[]){
Bank b;
b=new SBI();
System.out.println("Rate of Interest is: "+b.getRateOfInterest()+" %");
b=new PNB();
System.out.println("Rate of Interest is: "+b.getRateOfInterest()+" %");
}}

```

What is use of abstract class
Ans.abstract class provide the base of all subclass they can implement and extends and override the method use to implemented method in abstract class

48.Compare abstract class and interface
ans.

Abstract class	**Interface**

1) Abstract class can **have abstract and non-abstract** methods.	Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also.
2) Abstract class **doesn't support multiple inheritance**.	Interface **supports multiple inheritance**.
3) Abstract class **can have final, non-final, static and non-static variables**.	Interface has **only static and final variables**.
4) Abstract class **can provide the implementation of interface**.	Interface **can't provide the implementation of abstract class**.
5) The **abstract keyword** is used to declare abstract class.	The **interface keyword** is used to declare interface.
6) An **abstract class** can extend another Java class and implement multiple Java interfaces.	An **interface** can extend another Java interface only.
7) An **abstract class** can be extended using keyword "extends".	An **interface** can be implemented using keyword "implements".
8) A Java **abstract class** can have class members like private, protected, etc.	Members of a Java interface are public by default.

9)**Example:**
public abstract class Shape{
public abstract void draw();
}
	Example:
	public interface Drawable{
	void draw();
	}

```java
nterface A{
void a();//bydefault, public and abstract
void b();
void c();
void d();
}

//Creating abstract class that provides the implementation of one method of A interface
abstract class B implements A{
public void c(){System.out.println("I am C");}
}

//Creating subclass of abstract class, now we need to provide the implementation of rest of the methods
class M extends B{
public void a(){System.out.println("I am a");}
public void b(){System.out.println("I am b");}
public void d(){System.out.println("I am d");}
}

//Creating a test class that calls the methods of A interface
class Test5{
public static void main(String args[]){
```

```
A a=new M();
a.a();
a.b();
a.c();
a.d();
}}
```

49 what is inner class

Ans. **Java inner class** or nested class is a class that is declared inside the class or interface.

We use inner classes to logically group classes and interfaces in one place to be more readable and maintainable.

Additionally, it can access all the members of the outer class, including private data members and methods.

Waht is static inner class

ans. A static class is a class that is created inside a class, is called a static nested class in Java. It cannot access non-static data members and methods. It can be accessed by outer class name.

- It can access static data members of the outer class, including private.
- The static nested class cannot access non-static (instance) data members or
-

```
class TestOuter1{
  static int data=30;
  static class Inner{
   void msg(){System.out.println("data is "+data);}
  }
  public static void main(String args[]){
```

```
 TestOuter1.Inner obj=new TestOuter1.Inner();
 obj.msg();
 }
}
```

If you have the static member inside the static nested class, you don't need to create an instance of the static nested class.

```
public class TestOuter2{
  static int data=30;
  static class Inner{
   static void msg(){System.out.println("data is "+data);}
  }
  public static void main(String args[]){
  TestOuter2.Inner.msg();//no need to create the instance of static nested class
  }
}
```

51 can u create a inner class inside a method// local inner class

Ans. yes

71.diffrence between final finally and finalize

Ans final is the keyword

Final

Finally

Finalize

72.what is wrapper class in java

ans.A Wrapper class is a class whose object wraps or contains primitive data types. When we create an object to a wrapper class, it contains a field and in this field, we can store primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

73.what do you need wrapper classes

ans.Wrapper Class will convert primitive data types into objects. The objects are necessary if we wish to modify the arguments passed into the method (because primitive types are passed by value).

- The classes in java.util package handles only objects and hence wrapper classes help in this case also.
- Data structures in the Collection framework such as ArrayList and Vector store only the objects (reference types) and not the primitive types.
- The object is needed to support synchronization in multithreading.

74.what is string imutability

String references are used to store various attributes like username, password, etc. In Java, **String objects are immutable**. Immutable simply means unmodifiable or unchangeable.

Once String object is created its data or state can't be changed but a new String object is created.

75.where is string value store in memory

Ans.heap area

76 what is autoboxing

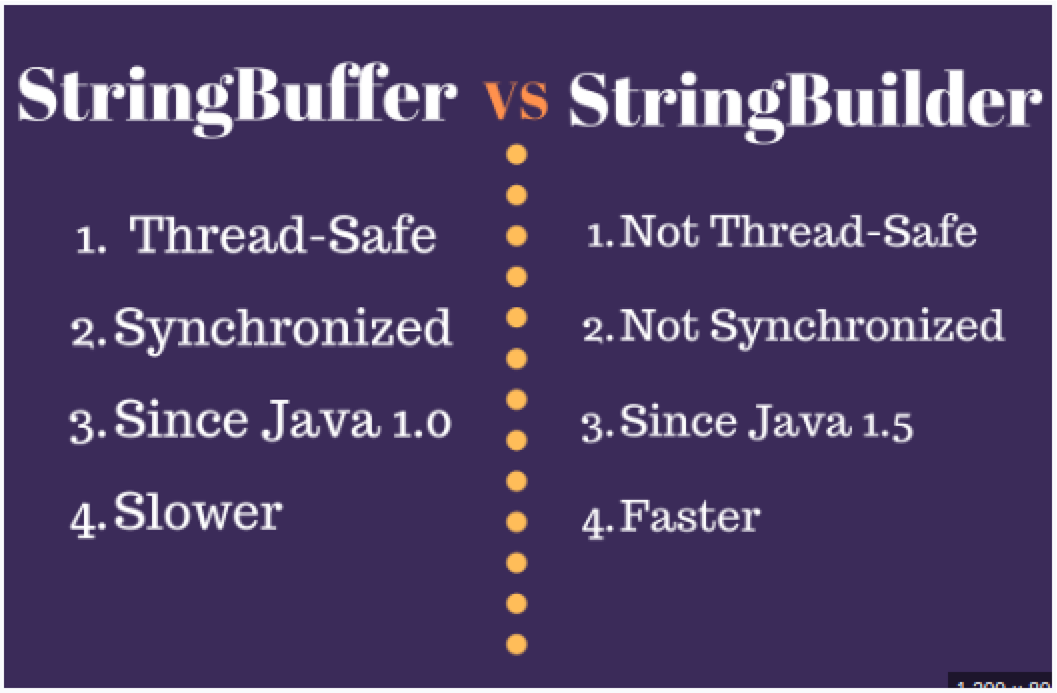
ans.Autoboxing is a procedure of converting a primitive value into an object of the corresponding wrapper class

77.what is unboxing

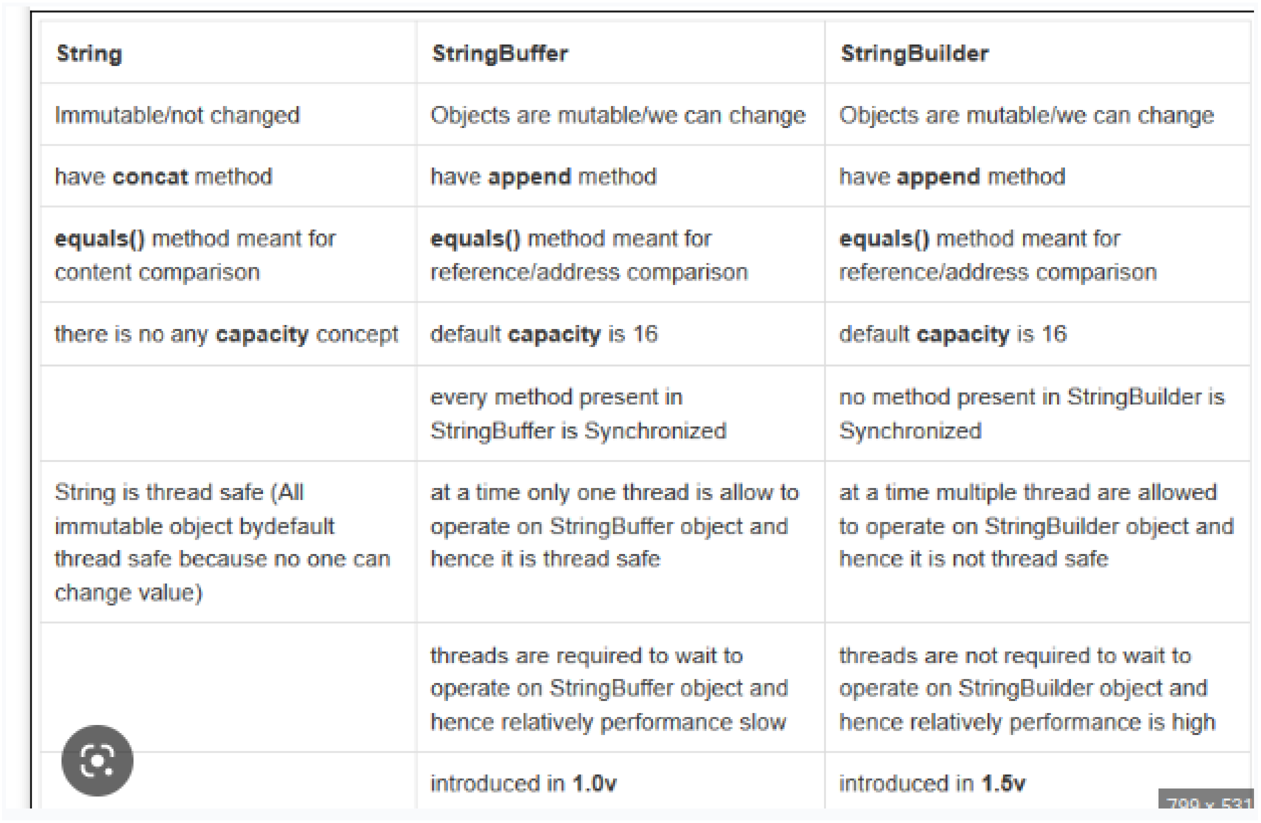
ans.**Unboxing** on the other hand refers to converting an object of a wrapper type to its corresponding primitive value.

78.difference between string ,stringbuilder,stringbuffer

ans.



StringBuffer	StringBuilder
1. Thread-Safe	1. Not Thread-Safe
2. Synchronized	2. Not Synchronized
3. Since Java 1.0	3. Since Java 1.5
4. Slower	4. Faster



String	StringBuffer	StringBuilder
Immutable/not changed	Objects are mutable/we can change	Objects are mutable/we can change
have **concat** method	have **append** method	have **append** method
equals() method meant for content comparison	**equals()** method meant for reference/address comparison	**equals()** method meant for reference/address comparison
there is no any **capacity** concept	default **capacity** is 16	default **capacity** is 16
	every method present in StringBuffer is Synchronized	no method present in StringBuilder is Synchronized
String is thread safe (All immutable object bydefault thread safe because no one can change value)	at a time only one thread is allow to operate on StringBuffer object and hence it is thread safe	at a time multiple thread are allowed to operate on StringBuilder object and hence it is not thread safe
	threads are required to wait to operate on StringBuffer object and hence relatively performance slow	threads are not required to wait to operate on StringBuilder object and hence relatively performance is high
	introduced in **1.0v**	introduced in **1.5v**

79.what is to string method

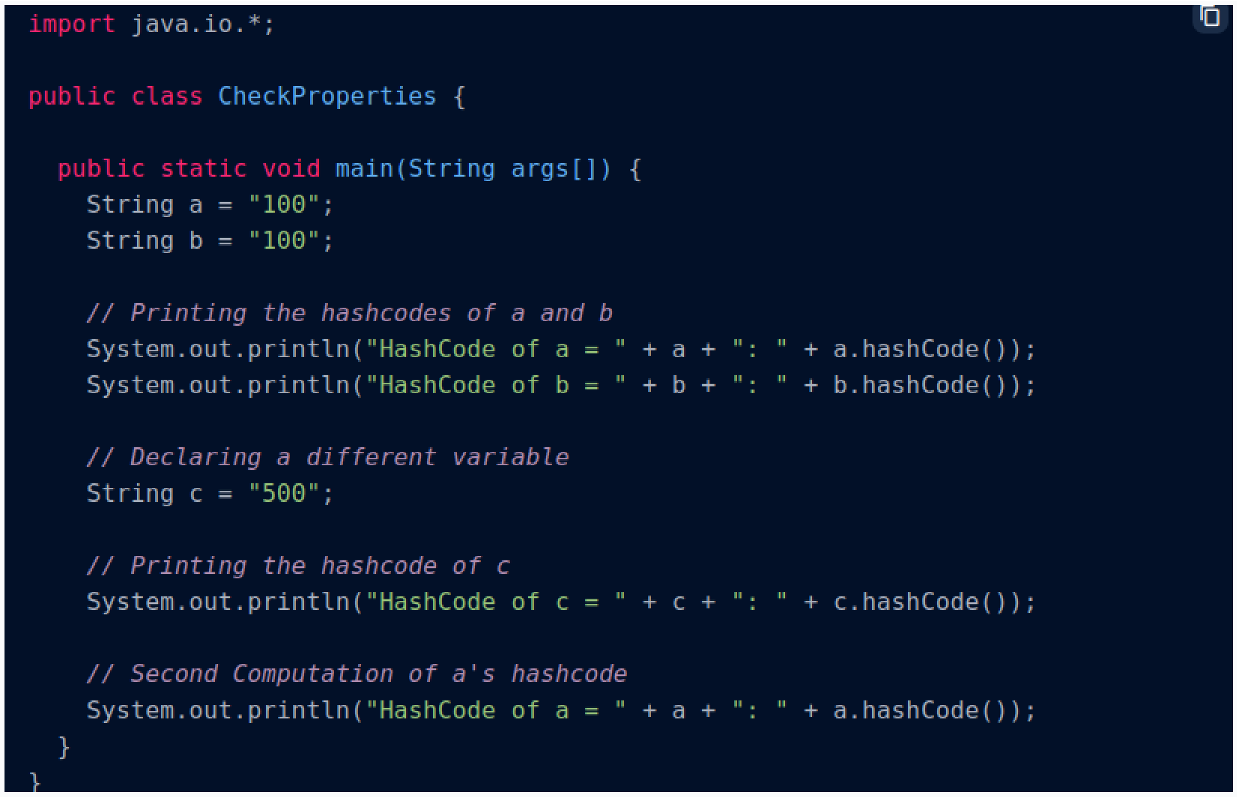
80. What is use of equal method in java

The **Java String class equals()** method compares the two given strings based on the content of the string. If any character is not matched, it returns false. If all characters are matched, it returns true.

The String equals() method overrides the equals() method of the Object class.

81.what is use of hashcode()method in java

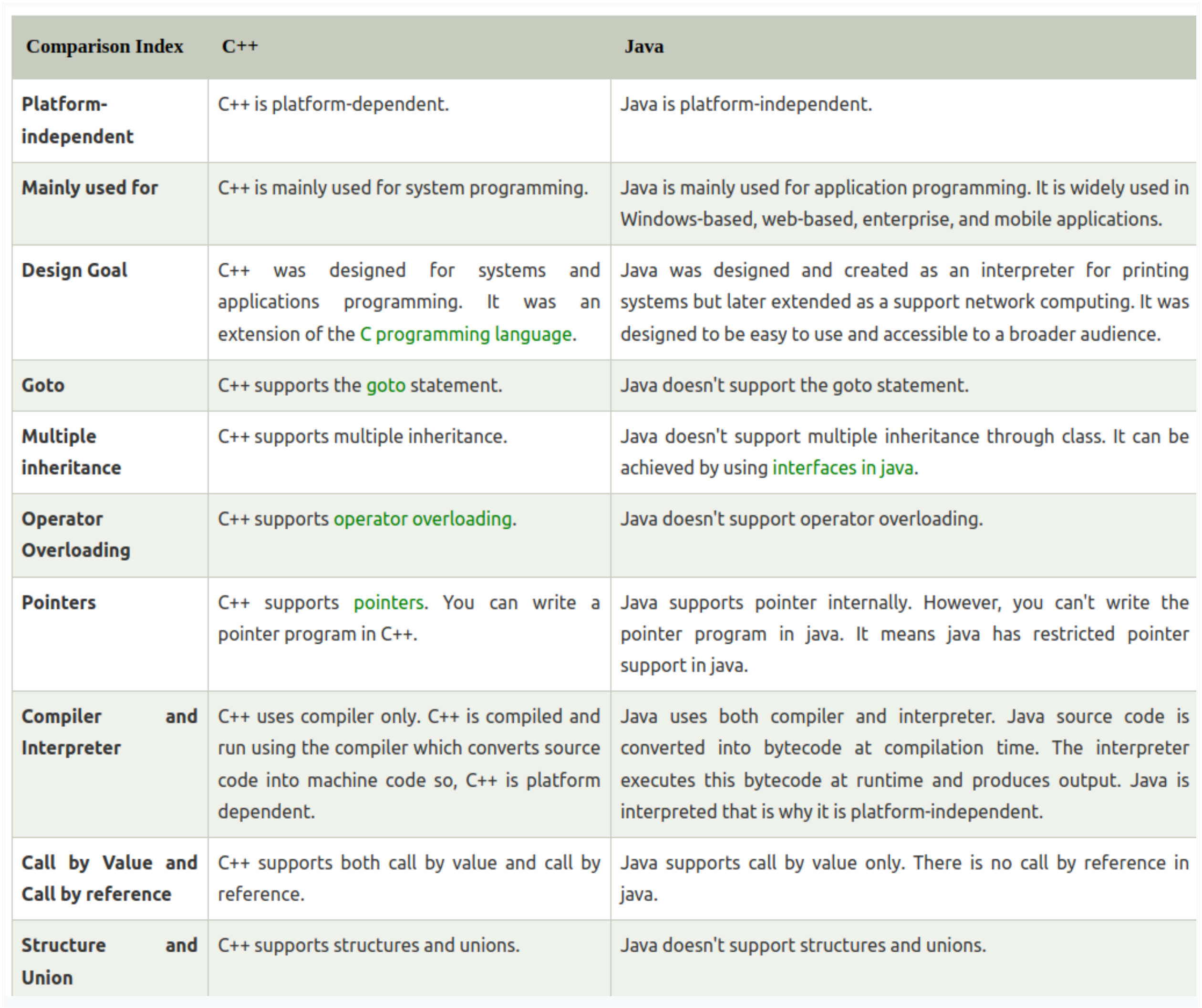
ansThe hashCode() method in Java is used to compute hash values of Java objects. The Integer class in Java contains two methods - hashCode() and hashCode(int value) which compute the hash values of Integer objects and primitive int values respectively.



```java
import java.io.*;

public class CheckProperties {

  public static void main(String args[]) {
    String a = "100";
    String b = "100";

    // Printing the hashcodes of a and b
    System.out.println("HashCode of a = " + a + ": " + a.hashCode());
    System.out.println("HashCode of b = " + b + ": " + b.hashCode());

    // Declaring a different variable
    String c = "500";

    // Printing the hashcode of c
    System.out.println("HashCode of c = " + c + ": " + c.hashCode());

    // Second Computation of a's hashcode
    System.out.println("HashCode of a = " + a + ": " + a.hashCode());
  }
}
```

82.how do you print content of array in java.

83.deffirence between in java ad c++



Comparison Index	C++	Java
Platform-independent	C++ is platform-dependent.	Java is platform-independent.
Mainly used for	C++ is mainly used for system programming.	Java is mainly used for application programming. It is widely used in Windows-based, web-based, enterprise, and mobile applications.
Design Goal	C++ was designed for systems and applications programming. It was an extension of the C programming language.	Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed to be easy to use and accessible to a broader audience.
Goto	C++ supports the goto statement.	Java doesn't support the goto statement.
Multiple inheritance	C++ supports multiple inheritance.	Java doesn't support multiple inheritance through class. It can be achieved by using interfaces in java.
Operator Overloading	C++ supports operator overloading.	Java doesn't support operator overloading.
Pointers	C++ supports pointers. You can write a pointer program in C++.	Java supports pointer internally. However, you can't write the pointer program in java. It means java has restricted pointer support in java.
Compiler and Interpreter	C++ uses compiler only. C++ is compiled and run using the compiler which converts source code into machine code so, C++ is platform dependent.	Java uses both compiler and interpreter. Java source code is converted into bytecode at compilation time. The interpreter executes this bytecode at runtime and produces output. Java is interpreted that is why it is platform-independent.
Call by Value and Call by reference	C++ supports both call by value and call by reference.	Java supports call by value only. There is no call by reference in java.
Structure and Union	C++ supports structures and unions.	Java doesn't support structures and unions.

84.what is classloader in java

ans.The **Java ClassLoader** is a part of the **Java Runtime Environment** that dynamically loads Java classes into the **Java Virtual Machine**.

85.what are different ways of creating wrappr class instances.

ans.Using the constructor of the wrapper class.

- Using the valueOf() method provided by the Wrapper classes.
- Using concept of AutoBoxing.

86.what is autoboxing

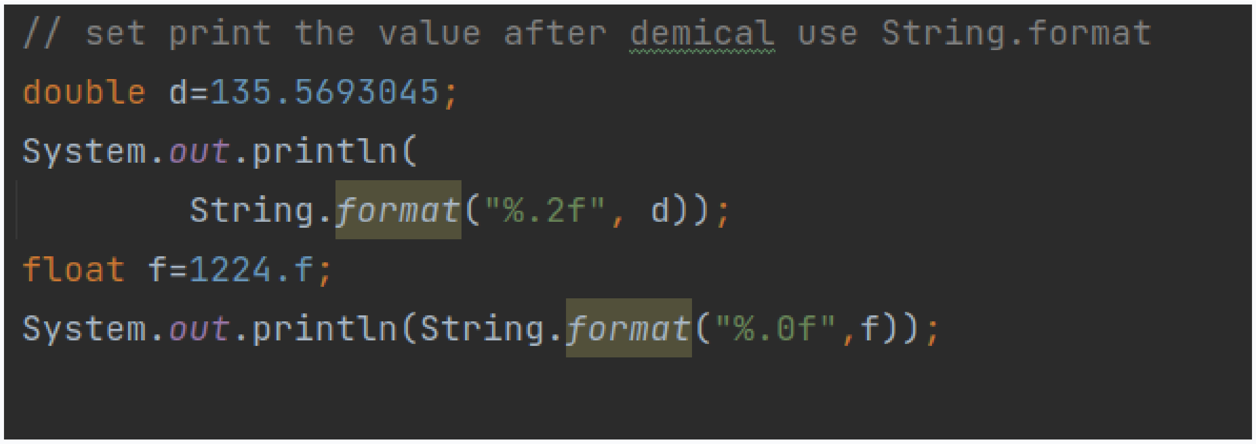
87.what are difference in two way creating a wrapper class in java.

ans.What are differences in the two ways of creating Wrapper Classes? The difference is that **using the Constructor you will always create a new object, while using valueOf() static method, it may return you a cached value with-in a range**.

89. What is string.format ,method

Ans we are print desire element after decimal

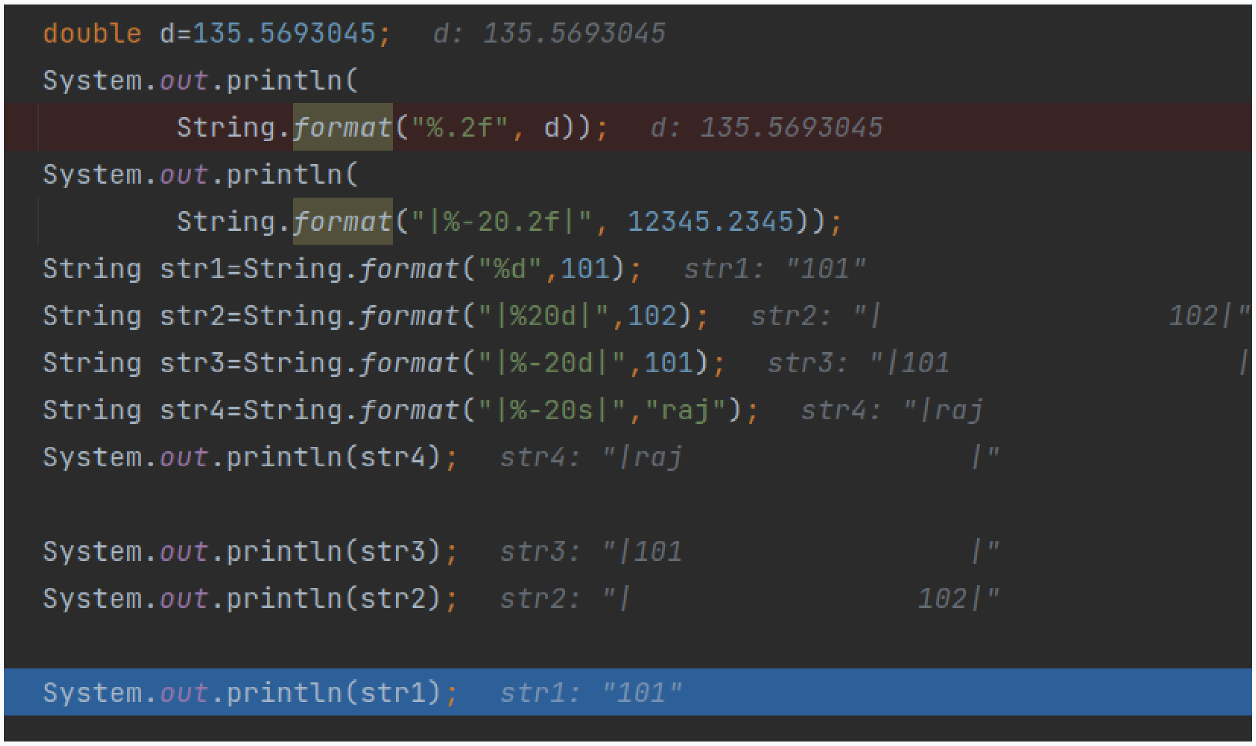
Use this method



```
// set print the value after demical use String.format
double d=135.5693045;
System.out.println(
        String.format("%.2f", d));
float f=1224.f;
System.out.println(String.format("%.0f",f));
```

90.String formatting in java

ans.



```
double d=135.5693045;   d: 135.5693045
System.out.println(
        String.format("%.2f", d));   d: 135.5693045
System.out.println(
        String.format("|%-20.2f|", 12345.2345));
String str1=String.format("%d",101);   str1: "101"
String str2=String.format("|%20d|",102);   str2: "|                 102|"
String str3=String.format("|%-20d|",101);   str3: "|101                 |
String str4=String.format("|%-20s|","raj");   str4: "|raj
System.out.println(str4);   str4: "|raj                 |"

System.out.println(str3);   str3: "|101                 |"
System.out.println(str2);   str2: "|                 102|"

System.out.println(str1);   str1: "101"
```