

```
class S2{  
    public static void main(String args[]){  
  
        StringBuffer sb1=new StringBuffer("hi");  
        StringBuffer sb2=new StringBuffer("hi");  
        System.out.println(sb1.equals(sb2)); //false  
  
    }  
  
}
```

Activate Windows
Go to Settings to activate Windows.

Ln 6, Col 45

90%

Windows (CRLF)

UTF-8

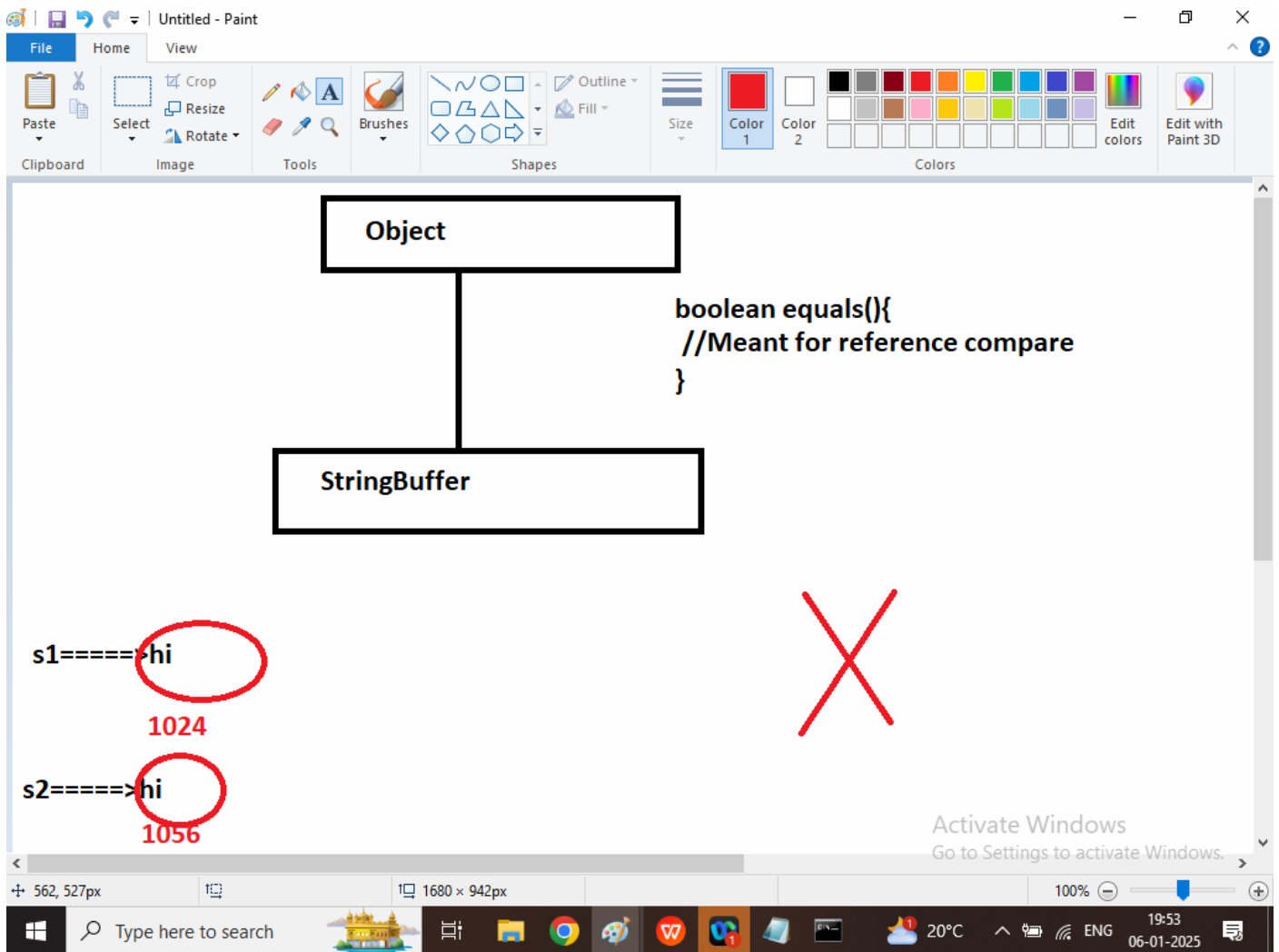


Type here to search



20°C

19:52
06-01-2025



```
class S2{  
    public static void main(String args[]){  
  
        String sb1=new String("hi");  
        String sb2=new String("hi");  
        System.out.println(sb1.equals(sb2)); //true  
  
    }  
}
```

Command Prompt

C:\Users\Admin\Desktop\Java Task>javac S2.java

C:\Users\Admin\Desktop\Java Task>java S2
true

C:\Users\Admin\Desktop\Java Task>

Activate Windows
Go to Settings to activate Windows.

Type here to search



20°C

19:53
06-01-2025

```
class S2{  
    public static void main(String args[]){  
  
        String sb1=new String("hi");  
        String sb2=new String("hi");  
        System.out.println(sb1.equals(sb2)); //true  
  
    }  
  
}
```

Activate Windows
Go to Settings to activate Windows.

Ln 6, Col 44

90%

Windows (CRLF)

UTF-8



Type here to search



20°C



19:54

06-01-2025

```
class S2{  
    public static void main(String args[]){  
  
        Object sb1=new String("hi");  
        Object sb2=new String("hi");  
        System.out.println(sb1.equals(sb2)); //true  
  
    }  
}
```

Command Prompt

```
C:\Users\Admin\Desktop\Java Task>java S2  
true
```

```
C:\Users\Admin\Desktop\Java Task>
```

Activate Windows
Go to Settings to activate Windows.

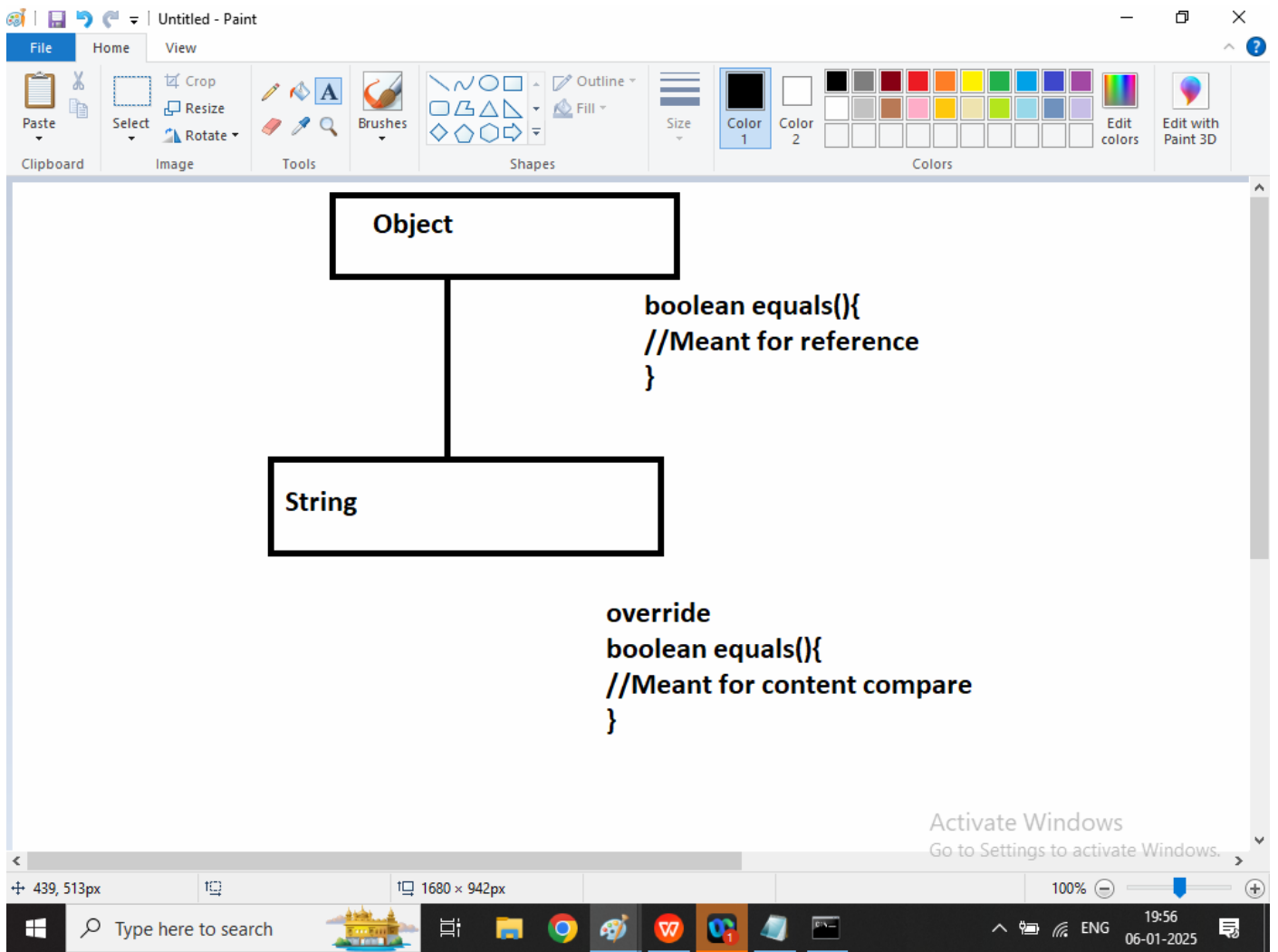


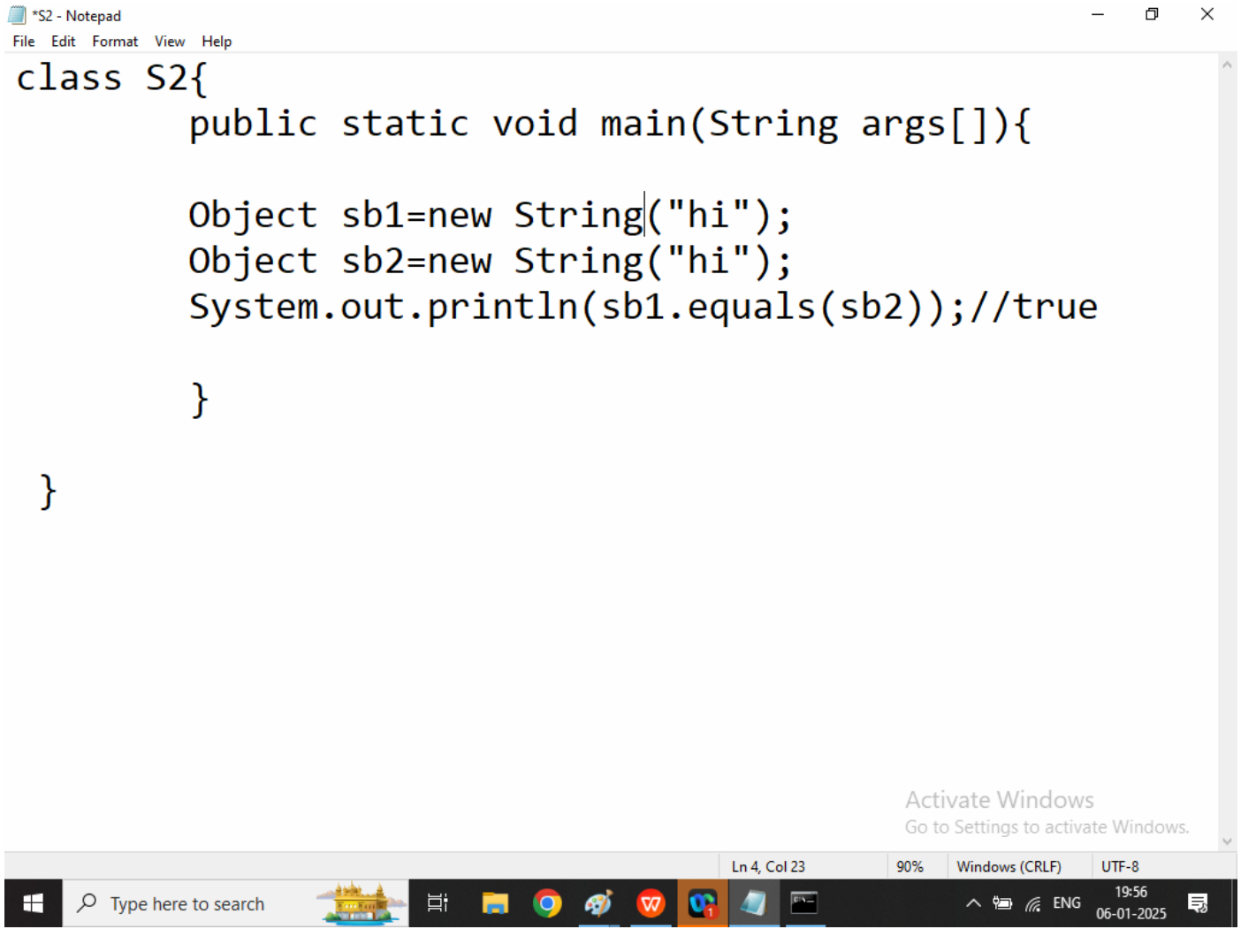
Type here to search



20°C

19:55
06-01-2025





```
class S2{  
    public static void main(String args[]){  
  
        Object sb1=new String("hi");  
        Object sb2=new String("hi");  
        System.out.println(sb1.equals(sb2)); //true  
  
    }  
  
}
```

Activate Windows
Go to Settings to activate Windows.

Ln 4, Col 23 90% Windows (CRLF) UTF-8
Type here to search 19:56 06-01-2025

Q2.Explain Method Overriding in java Programming?

Ans: Method overriding in java is a features that allows a sub class to provide a specific implementation of a method already defined in parent class. It is used to achieve run time Polymorphism in java and customize or enhance behaviour of an inherited method

Some points about method overriding

1. A class must have a “IS-A” Relationship
2. Method name and Signature must be same as its parent class method
3. Inheritance: The sub class must inherit from the super class where method is defined

```
class Parent{
    public void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method....");
    }

    public static void main(String args[]){
        Parent c=new Parent();
        c.show();
    }
}
```

Command Prompt

```
C:\Users\Admin\Desktop\Java Task>java Child
This is Parent class show method....
```

```
C:\Users\Admin\Desktop\Java Task>
Activate Windows
Go to Settings to activate Windows.
```



Type here to search



Earn...

20:07
06-01-2025


```
class Parent{
    public void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method....");
    }

    public static void main(String args[]){
        Parent c=new Child();
        c.show();
    }
}
```

```
C:\Users\Admin\Desktop>
This is Parent class

C:\Users\Admin\Desktop>
This is Child class s

C:\Users\Admin\Desktop>
This is Child class s

C:\Users\Admin\Desktop>
This is Child class s
```

Activate Windows

Go to Settings to activate Windows.

Ln 13, Col 20



Type here to search

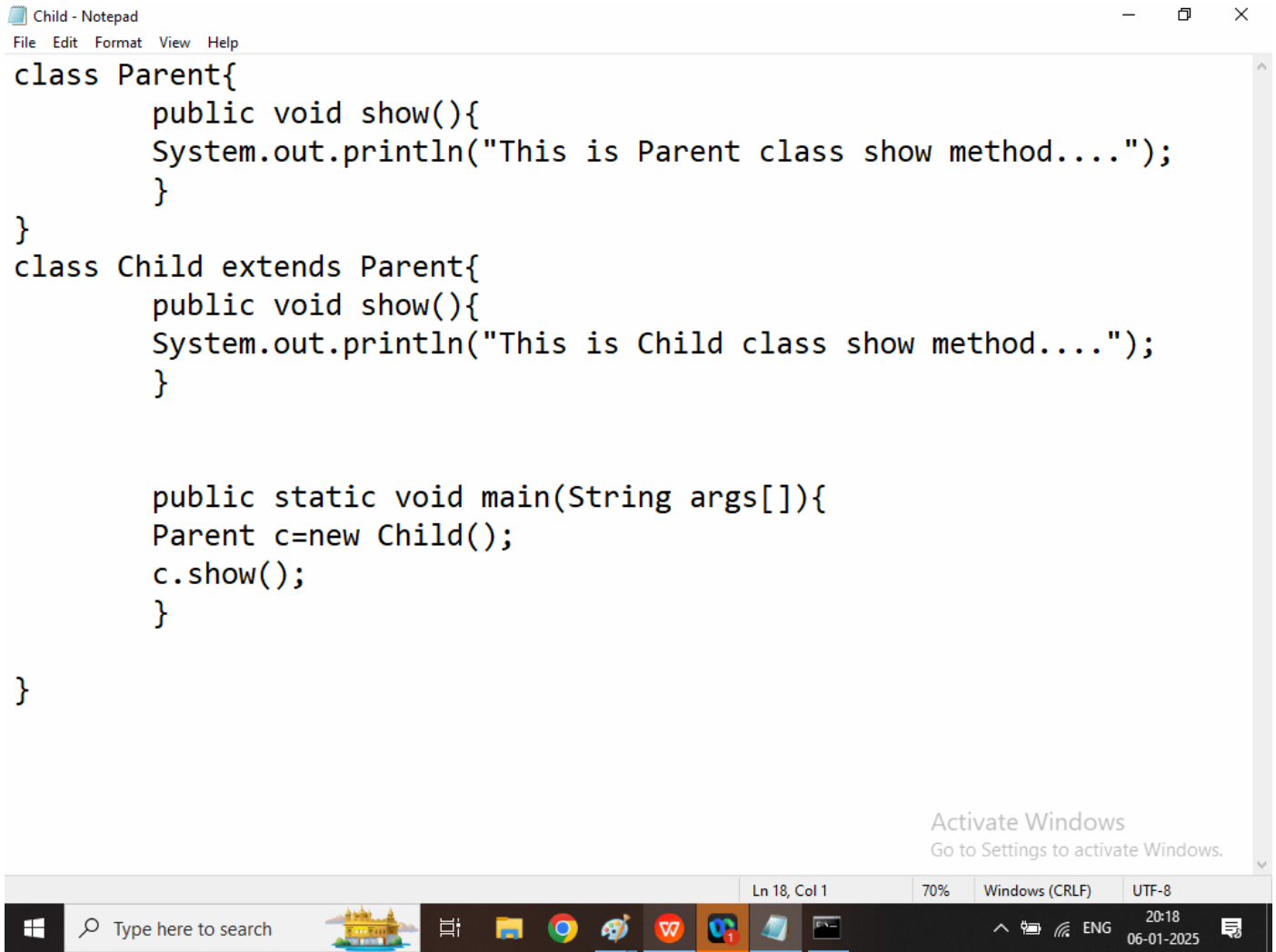


Earn...



ENG

20:08
06-01-2025



The screenshot shows a Windows Notepad window titled "Child - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The code is as follows:

```
class Parent{
    public void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method....");
    }

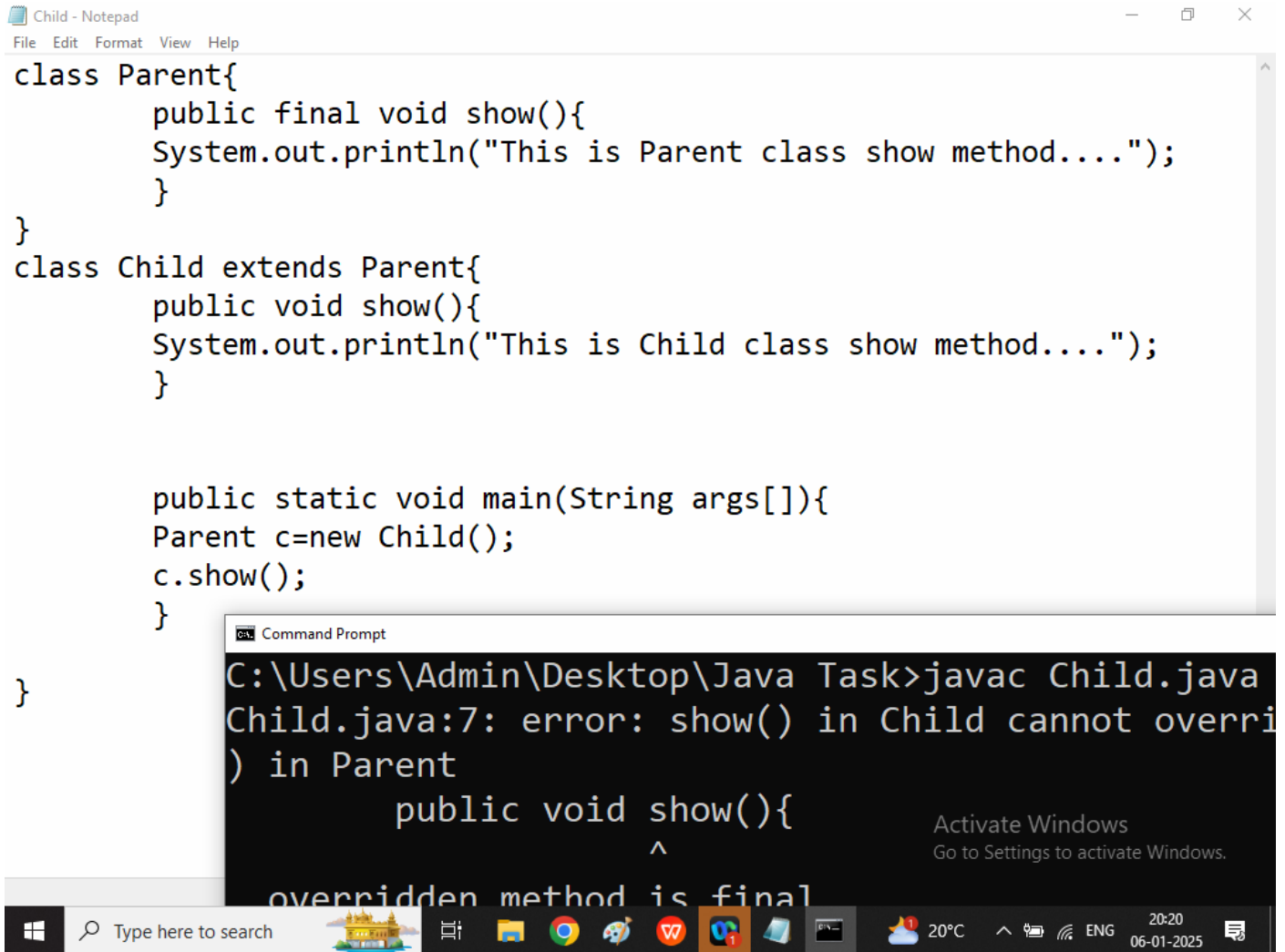
    public static void main(String args[]){
        Parent c=new Child();
        c.show();
    }
}
```

At the bottom right of the Notepad window, there is a watermark that says "Activate Windows" and "Go to Settings to activate Windows." The status bar at the bottom of the window shows "Ln 18, Col 1", "70%", "Windows (CRLF)", "UTF-8", and the system clock "20:18 06-01-2025". The taskbar at the very bottom contains the Start button, a search bar, and several application icons including File Explorer, Google Chrome, and Microsoft Word.

Q1. Explain final keyword in java?

Ans: It is used to perform some restrictions

1. Final variable: cannot be changes
2. Final method: cannot be override



The screenshot shows a Windows desktop environment. In the foreground, a Notepad window titled 'Child - Notepad' contains the following Java code:

```
class Parent{
    public final void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method....");
    }

    public static void main(String args[]){
        Parent c=new Child();
        c.show();
    }
}
```

Below the Notepad window, a Command Prompt window is open, displaying the command to compile the code and the resulting error:

```
C:\Users\Admin\Desktop\Java Task>javac Child.java
Child.java:7: error: show() in Child cannot override
    ) in Parent
        public void show(){
                ^
    overridden method is final
```

The error message indicates that the `show()` method in the `Child` class cannot override the `show()` method in the `Parent` class because the `show()` method in the `Parent` class is `final`.

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right displays the temperature (20°C), network status, and the date and time (20:20, 06-01-2025).

3. Final class: final class cannot be inherit

```
class Parent{
    final int MAX=10;
    public void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method...."+MAX);
    }

    public static void main(String args[]){
        Parent c=new Child();
        c.show();
    }
}
```

Activate Windows
Go to Settings to activate Windows.

Ln 4, Col 2

70%

Windows (CRLF)

UTF-8



Type here to search



20°C



20:23

06-01-2025

```
final class Parent{
    public void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method....");
    }

    public static void main(String args[]){
        Parent c=new Child();
        c.show();
    }
}
```

Select Command Prompt

```
Child.java:6: error: cannot inherit from final Par
class Child extends Parent{
    ^
```

1 error

Activate Windows
Go to Settings to activate Windows.

C:\Users\Admin\Desktop\Java Task>

```
class Parent{
    final int MAX=10;
    public void show(){
        System.out.println("This is Parent class show method....");
    }
}
class Child extends Parent{
    public void show(){
        System.out.println("This is Child class show method...."+super.MAX);
        final int MAX=55;
        System.out.println("This is Child class show method...."+MAX);
    }

    public static void main(String args[]){

        Parent c=new Child();
        c.show();
    }
}
```

Command Prompt

C:\Users\Admin\Desktop\Java Task>javac

C:\Users\Admin\Desktop\Java Task>java C

This is Child class show method....10

This is Child class show method....55

Activate Windows

Go to Settings to activate Windows.

C:\Users\Admin\Desktop\Java Task>



Type here to search



20:27

06-01-2025