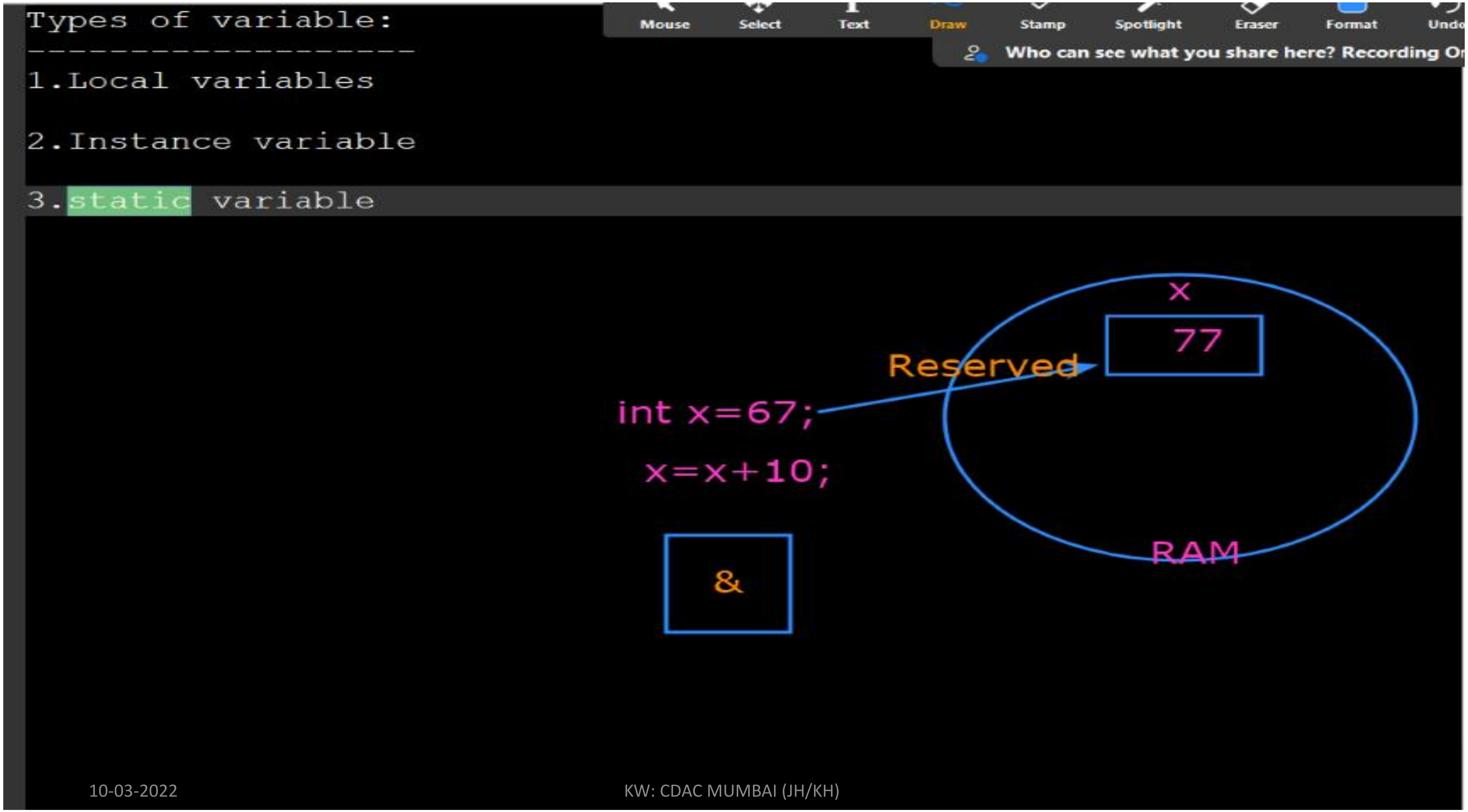
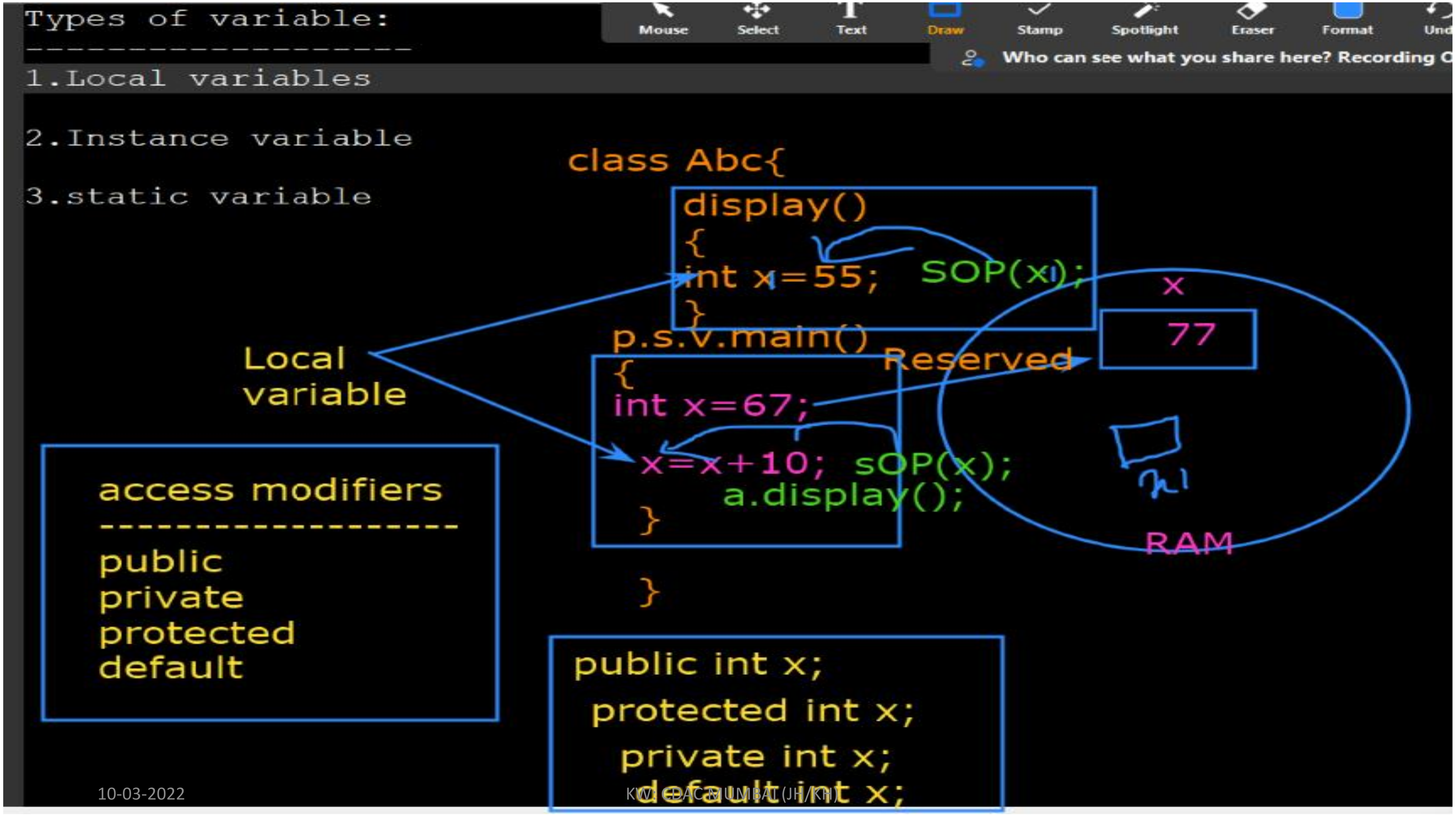
Concepts of Programming
Day 3: March 2022

Kiran Waghmare



```
Types of variable:
---------------------
1.Local variables

2.Instance variable

3.static variable
```



Types of variable:

1.Local variables

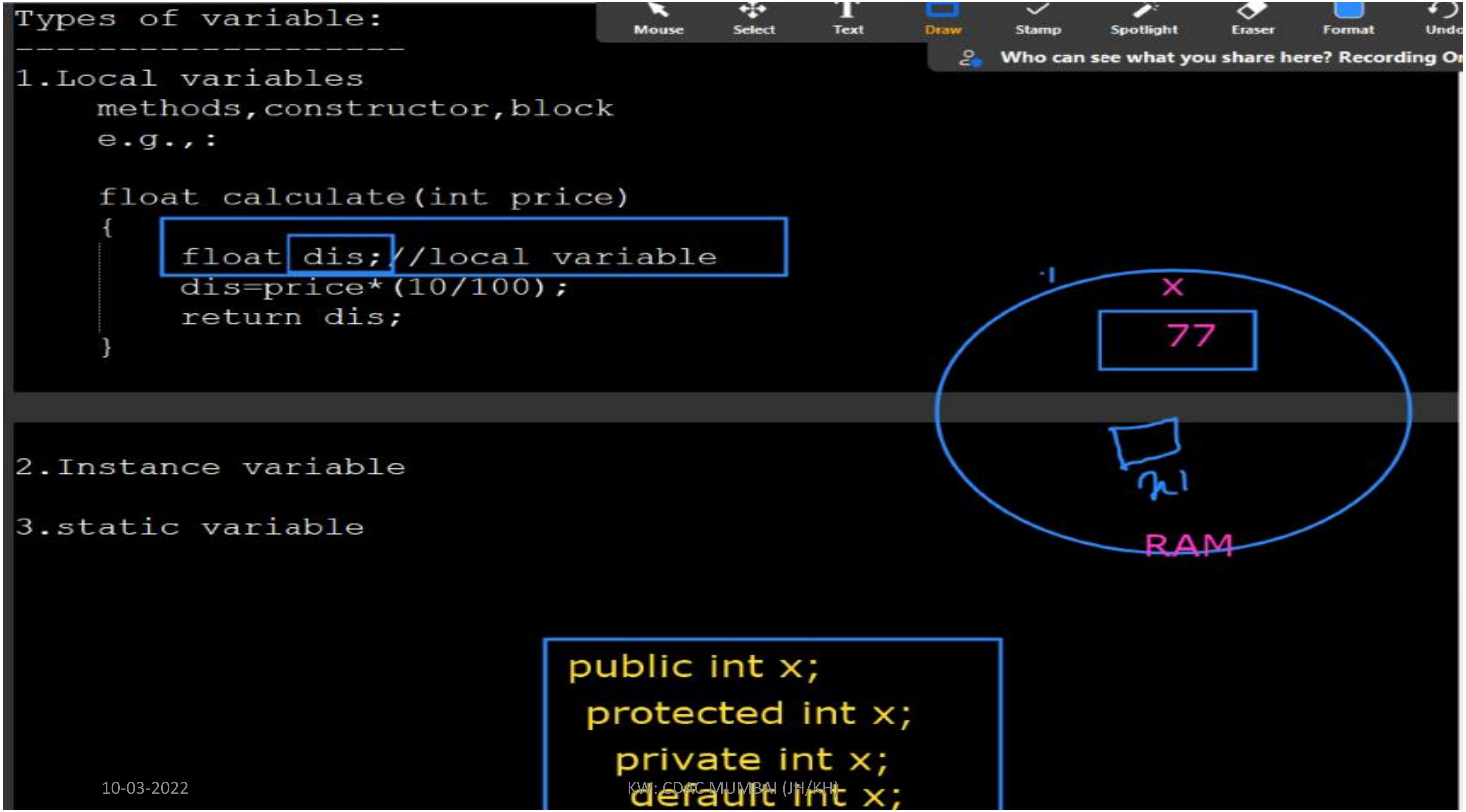
2.Instance variable

3.static variable

```
class Abc{
display()
{
int x=55;   SOP(x);
}
p.s.v.main()
{
int x=67;
x=x+10;  sOP(x);
    a.display();
}

}
```

access modifiers
----
public
private
protected
default

```
public int x;
 protected int x;
  private int x;
  default int x;
```



Who can see what you share here? Recording On

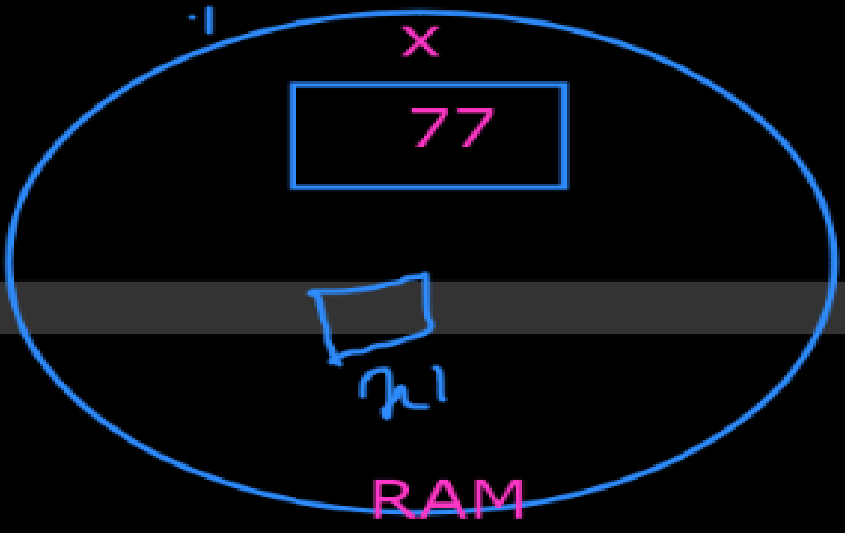
```
Types of variable:
-----------------------
1.Local variables
    methods,constructor,block
    e.g.,:

    float calculate(int price)
    {
        float dis;//local variable
        dis=price*(10/100);
        return dis;
    }


2.Instance variable

3.static variable
```

```
public int x;
protected int x;
private int x;
default int x;
```

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo

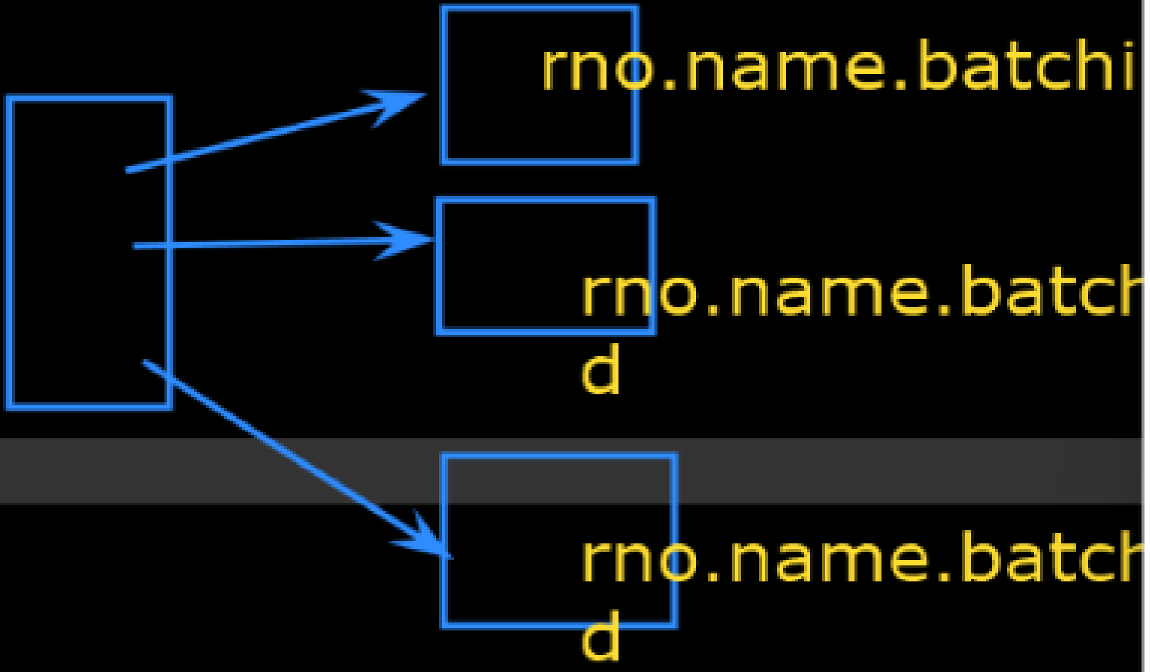
Who can see what you share here? Recording On

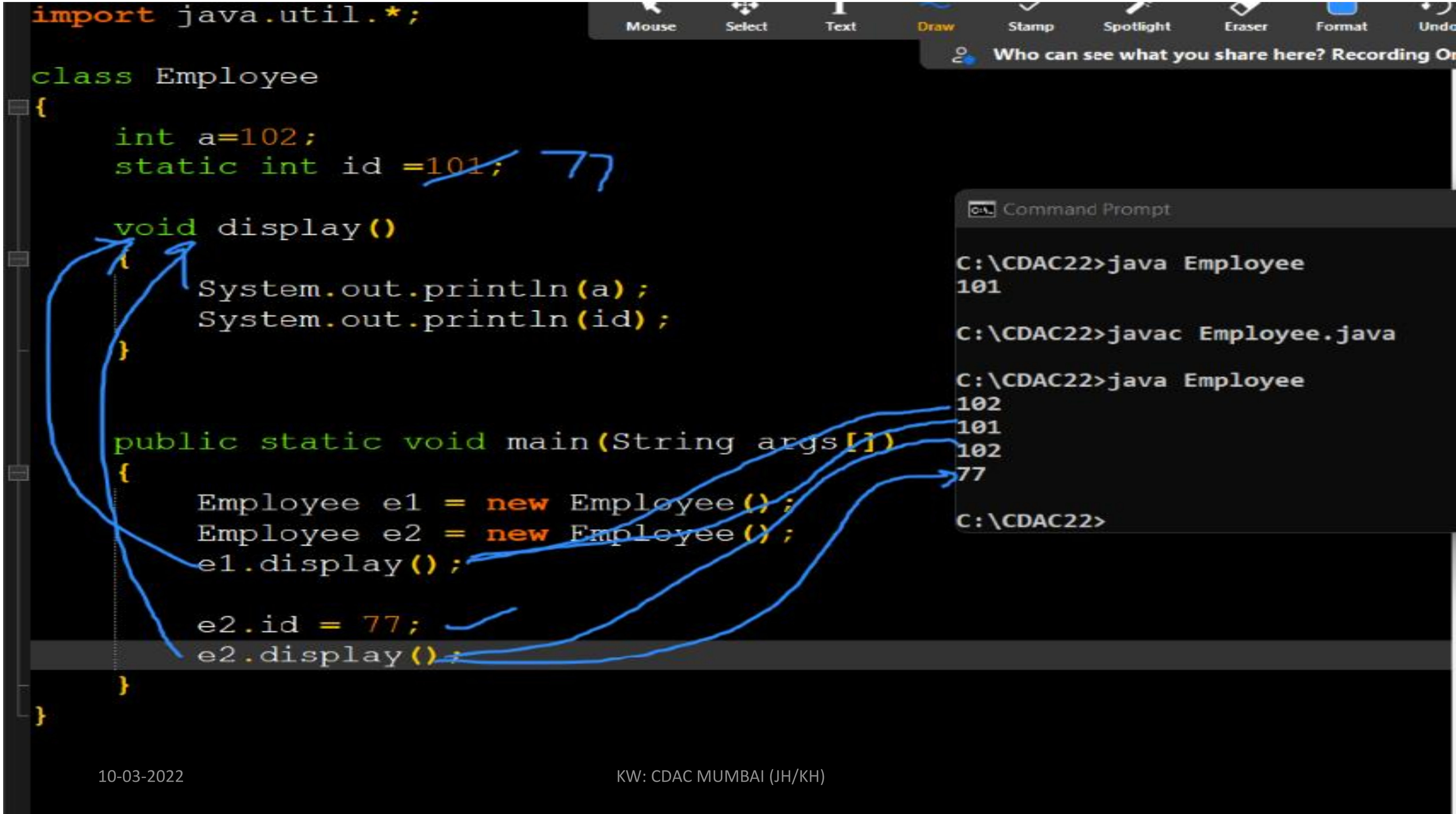
```
Types of variable:
--------------------
1.Local variables
    methods,constructor,block
    e.g.,:

    float calculate(int price)
    {
        float dis;//local variable
        dis=price*(10/100);
        return dis;
    }

2.Instance variable

3.static variable
```



public int x;
protected int x;
private int x;
default int x;

```
3.static variable
-static variables are initialized only once.
-declaring constant + final
-static variable = instance variable
-static means constant

class student
{
int Rollno;
String name;
static String batchid="Sep2022";
}
```

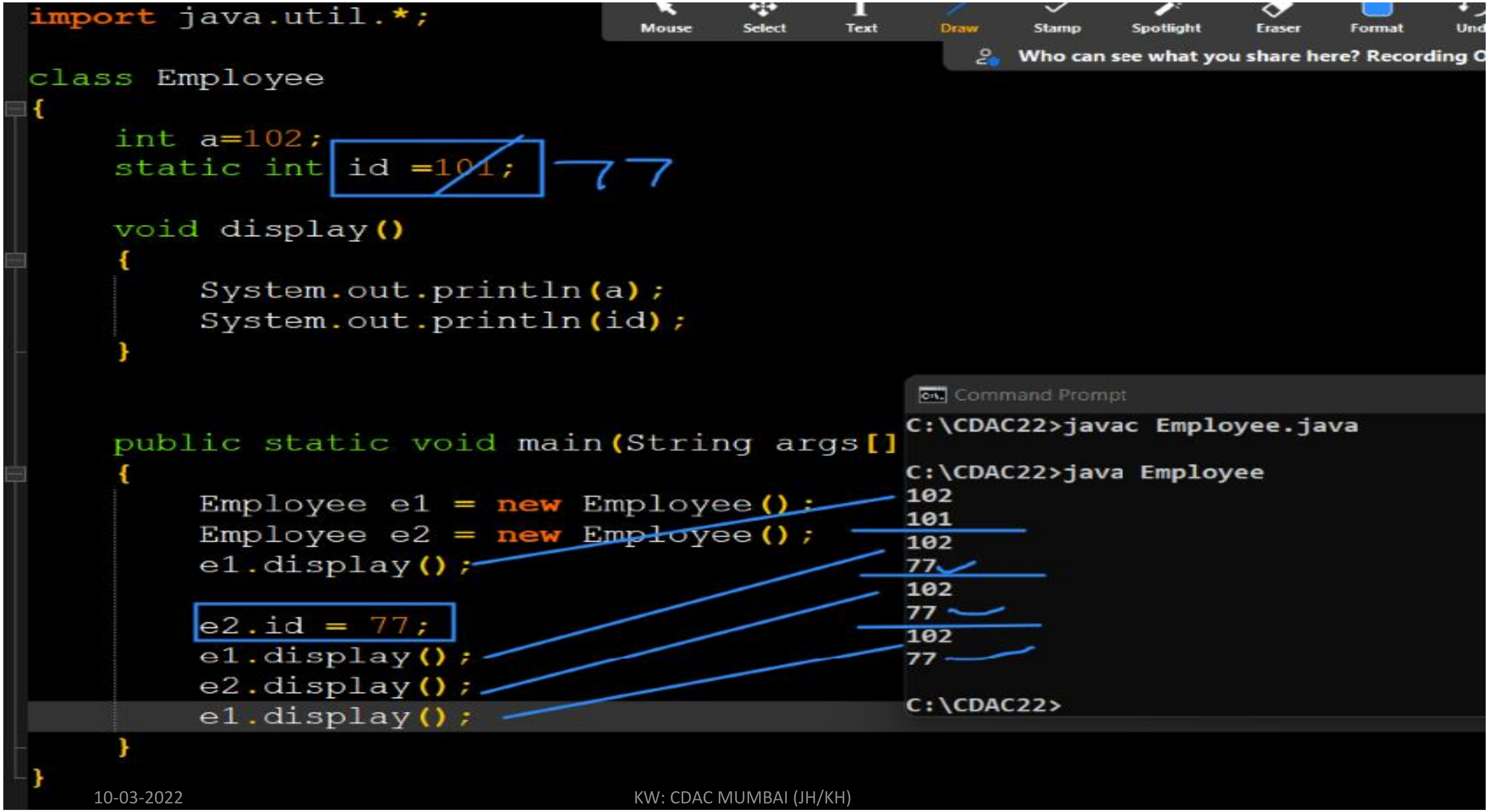




```
import java.util.*;

class Employee
{
    int a=102;
    static int id =~~101;~~  77

    void display()
    {
        System.out.println(a);
        System.out.println(id);
    }


    public static void main(String args[])
    {
        Employee e1 = new Employee();
        Employee e2 = new Employee();
        e1.display();

        e2.id = 77;
        e2.display();
    }
}
```

Command Prompt

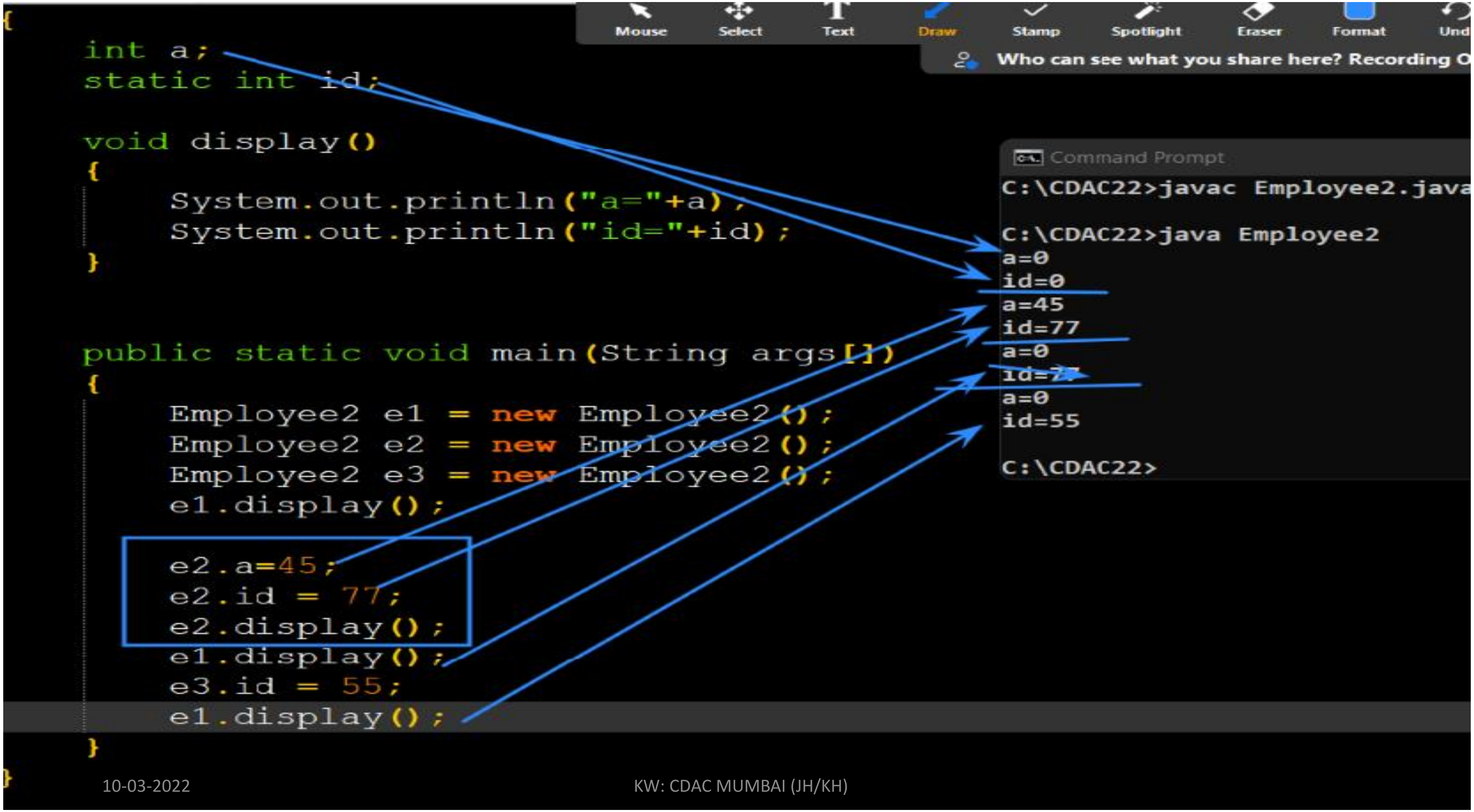
```
C:\CDAC22>java Employee
101

C:\CDAC22>javac Employee.java

C:\CDAC22>java Employee
102
101
102
77

C:\CDAC22>
```



```java
import java.util.*;

class Employee
{
    int a=102;
    static int id =~~101~~;

    void display()
    {
        System.out.println(a);
        System.out.println(id);
    }


    public static void main(String args[]
    {
        Employee e1 = new Employee();
        Employee e2 = new Employee();
        e1.display();

        e2.id = 77;
        e1.display();
        e2.display();
        e1.display();
    }
}
```

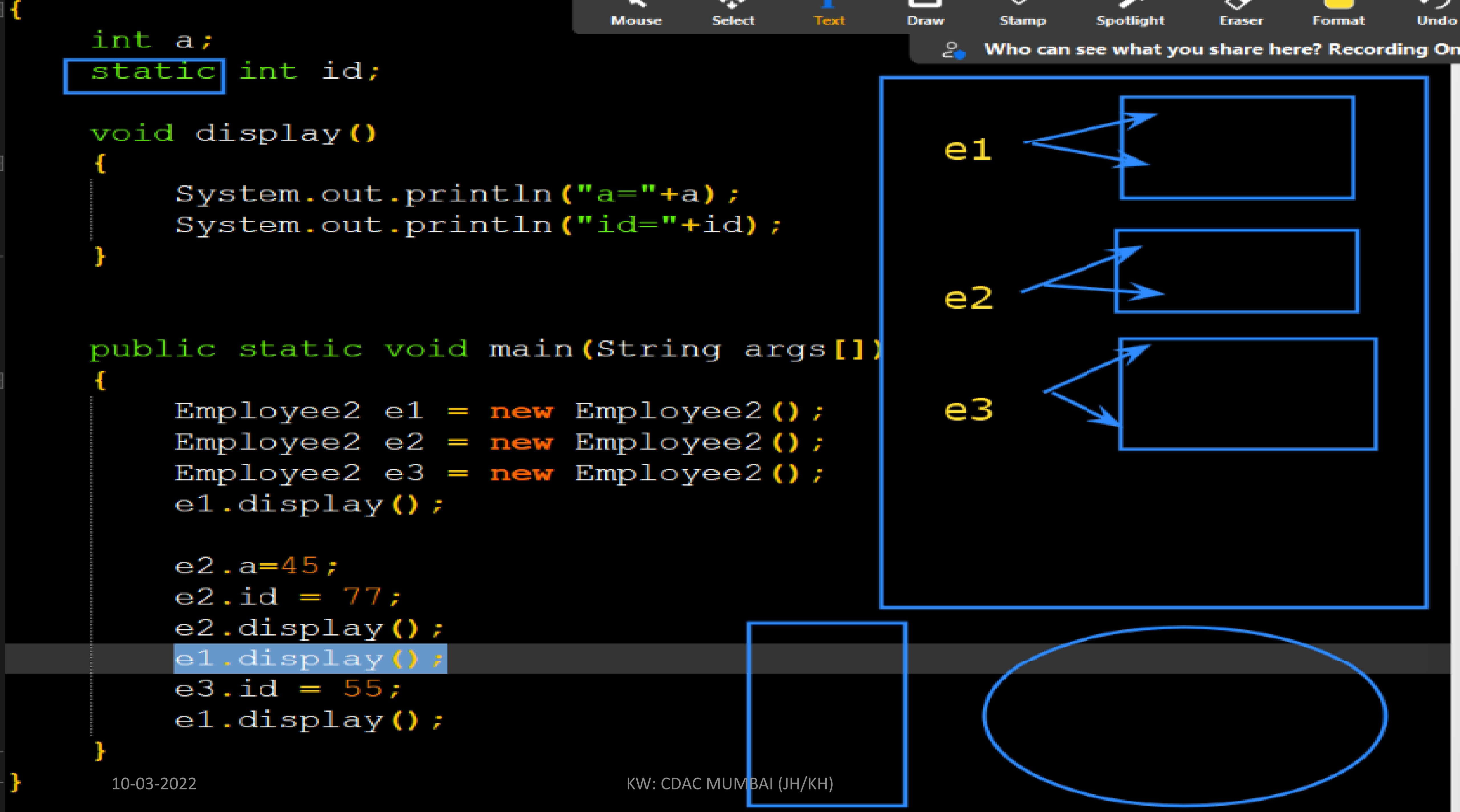
Command Prompt

```
C:\CDAC22>javac Employee.java

C:\CDAC22>java Employee
102
101
102
77
102
77
102
77

C:\CDAC22>
```

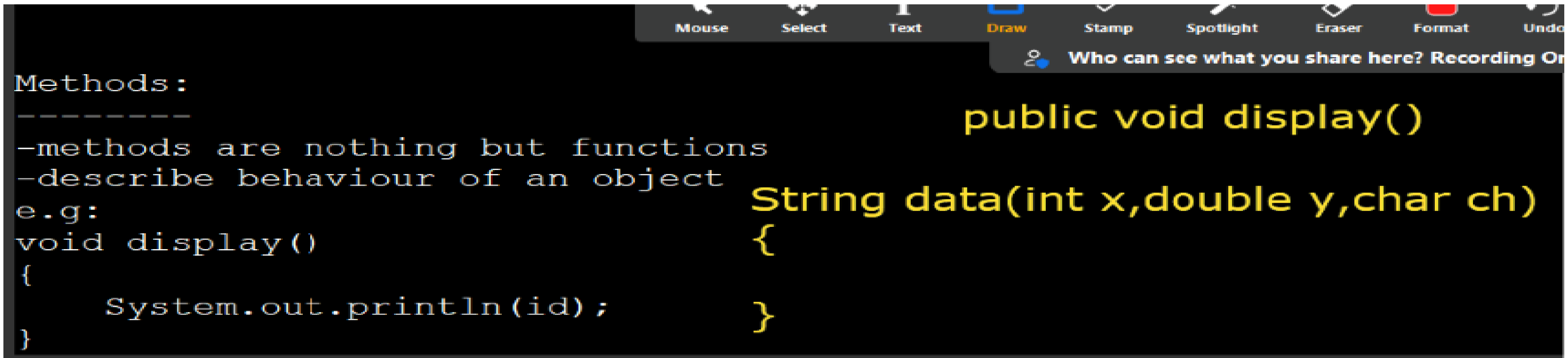


```
{
    int a;
    static int id;

    void display()
    {
        System.out.println("a="+a);
        System.out.println("id="+id);
    }


    public static void main(String args[])
    {
        Employee2 e1 = new Employee2();
        Employee2 e2 = new Employee2();
        Employee2 e3 = new Employee2();
        e1.display();

        e2.a=45;
        e2.id = 77;
        e2.display();
        e1.display();
        e3.id = 55;
        e1.display();
    }
}
```

```
C:\CDAC22>javac Employee2.java

C:\CDAC22>java Employee2
a=0
id=0
a=45
id=77
a=0
id=77
a=0
id=55

C:\CDAC22>
```



```
{
    int a;
    static int id;

    void display()
    {
        System.out.println("a="+a);
        System.out.println("id="+id);
    }


    public static void main(String args[])
    {
        Employee2 e1 = new Employee2();
        Employee2 e2 = new Employee2();
        Employee2 e3 = new Employee2();
        e1.display();

        e2.a=45;
        e2.id = 77;
        e2.display();
        e1.display();
        e3.id = 55;
        e1.display();
    }
}
```



```
Methods:
---------
-methods are nothing but functions
-describe behaviour of an object
e.g:
void display()
{
     System.out.println(id);
}
```

```
public void display()

String data(int x,double y,char ch)
{

}
```

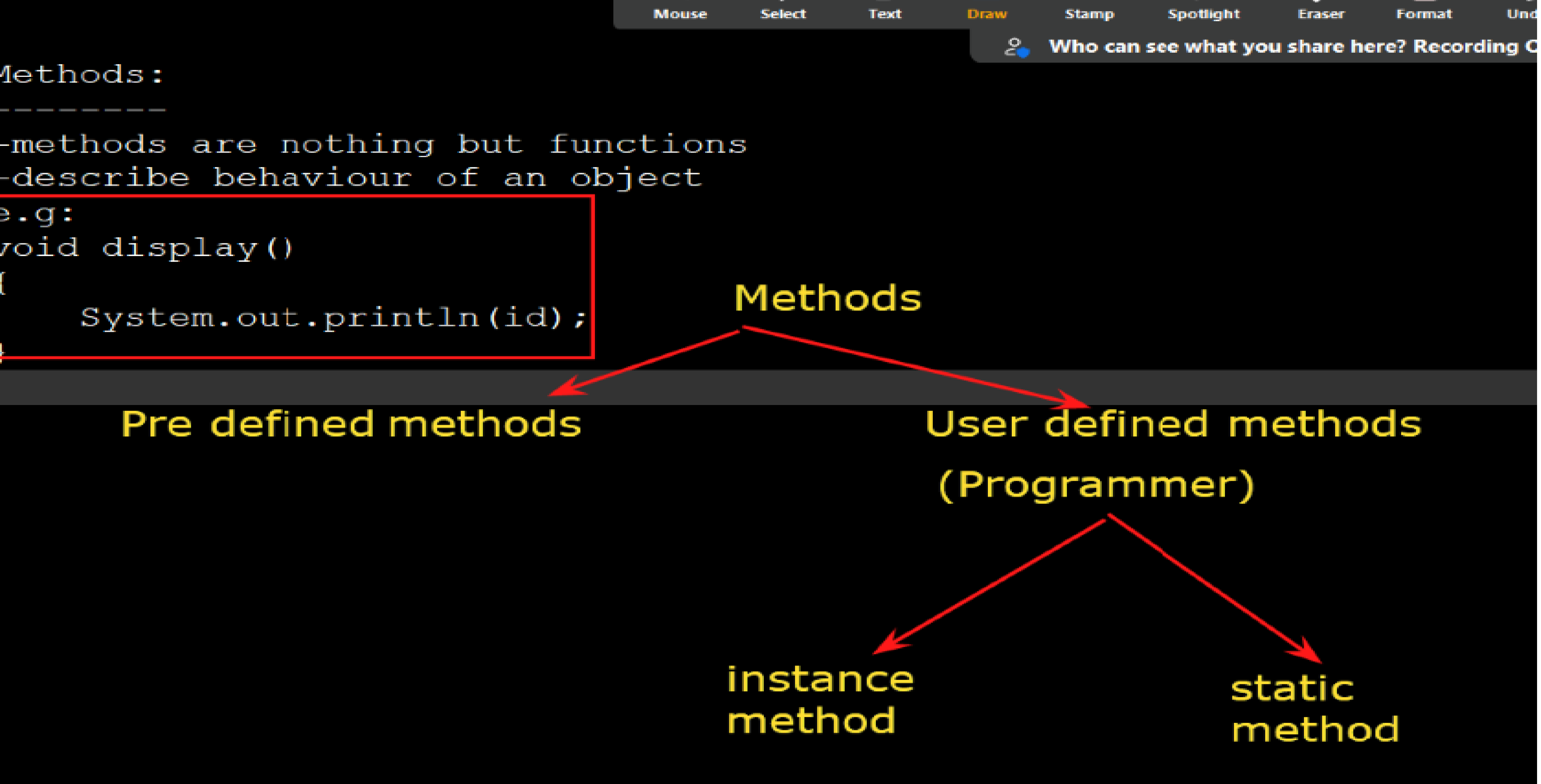
```
modifier    Return-type    method name(parameter)
{
        //method body
}
```

access modifier: public,private,protected,default

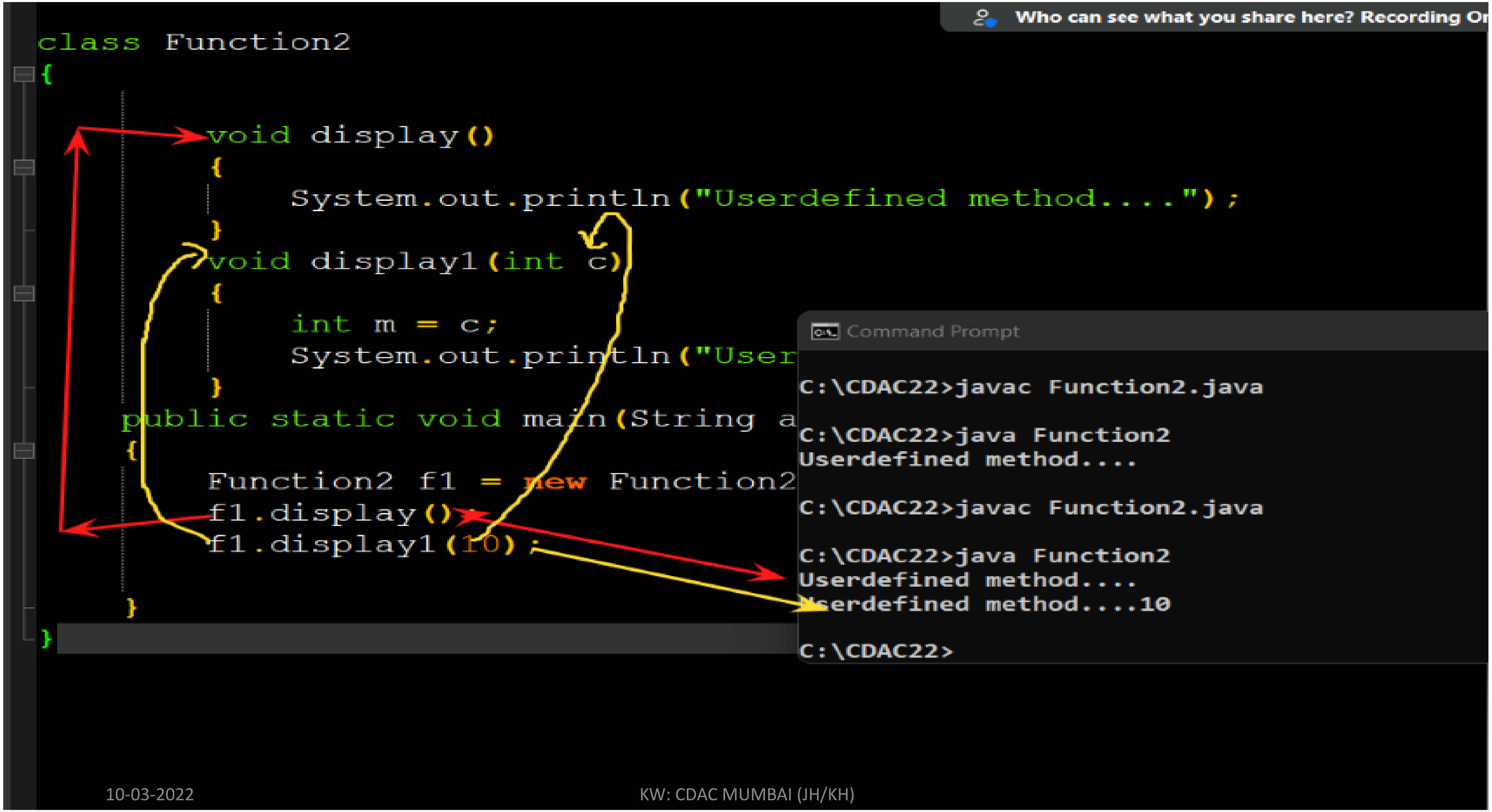
return-type: method may have return value (data type)

method-name: userdefine

parameter: value passed to a method



```
Methods:
---------
-methods are nothing but functions
-describe behaviour of an object
e.g:
void display()
{
    System.out.println(id);
}
```

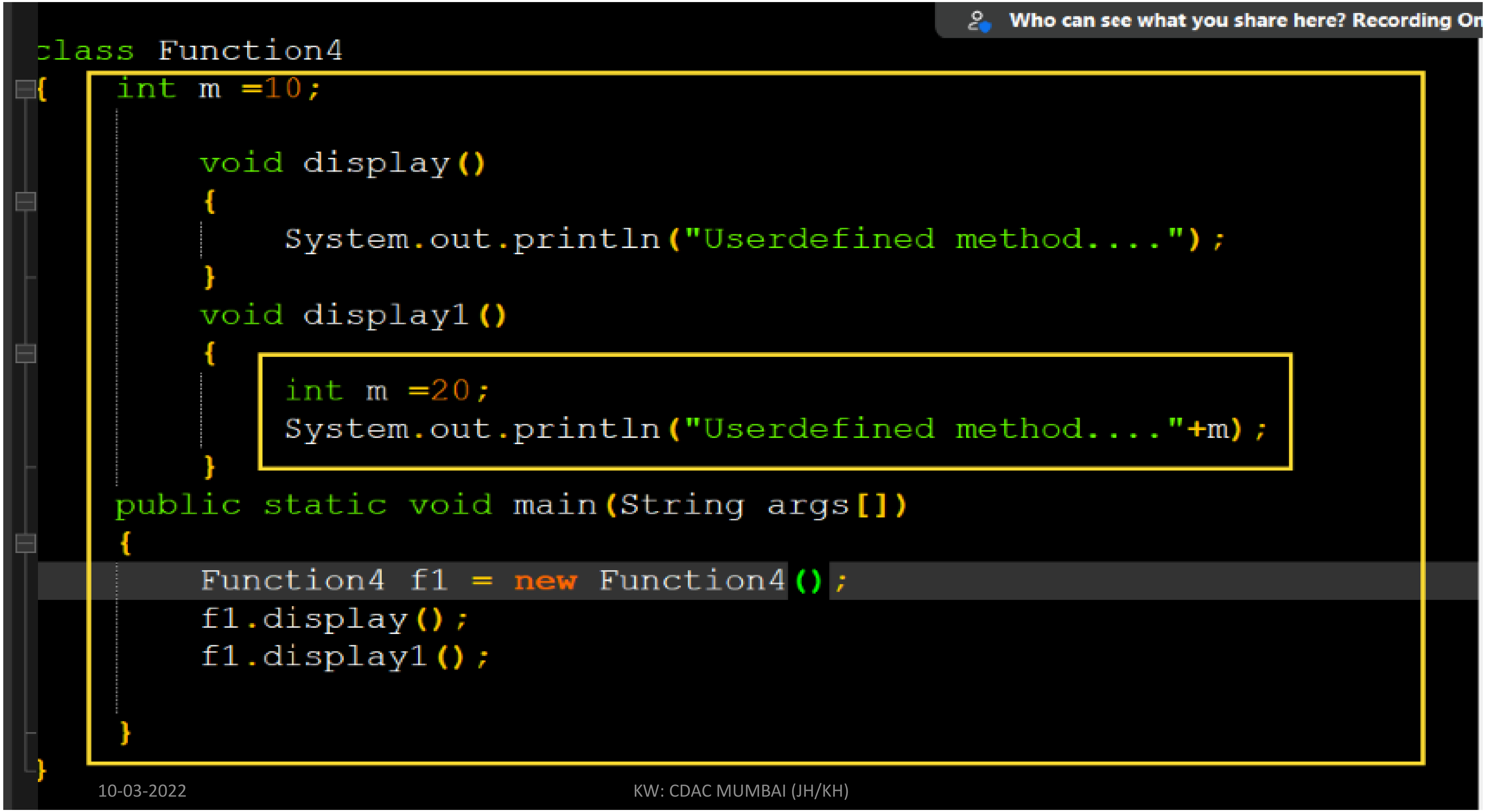


Who can see what you share here? Recording On

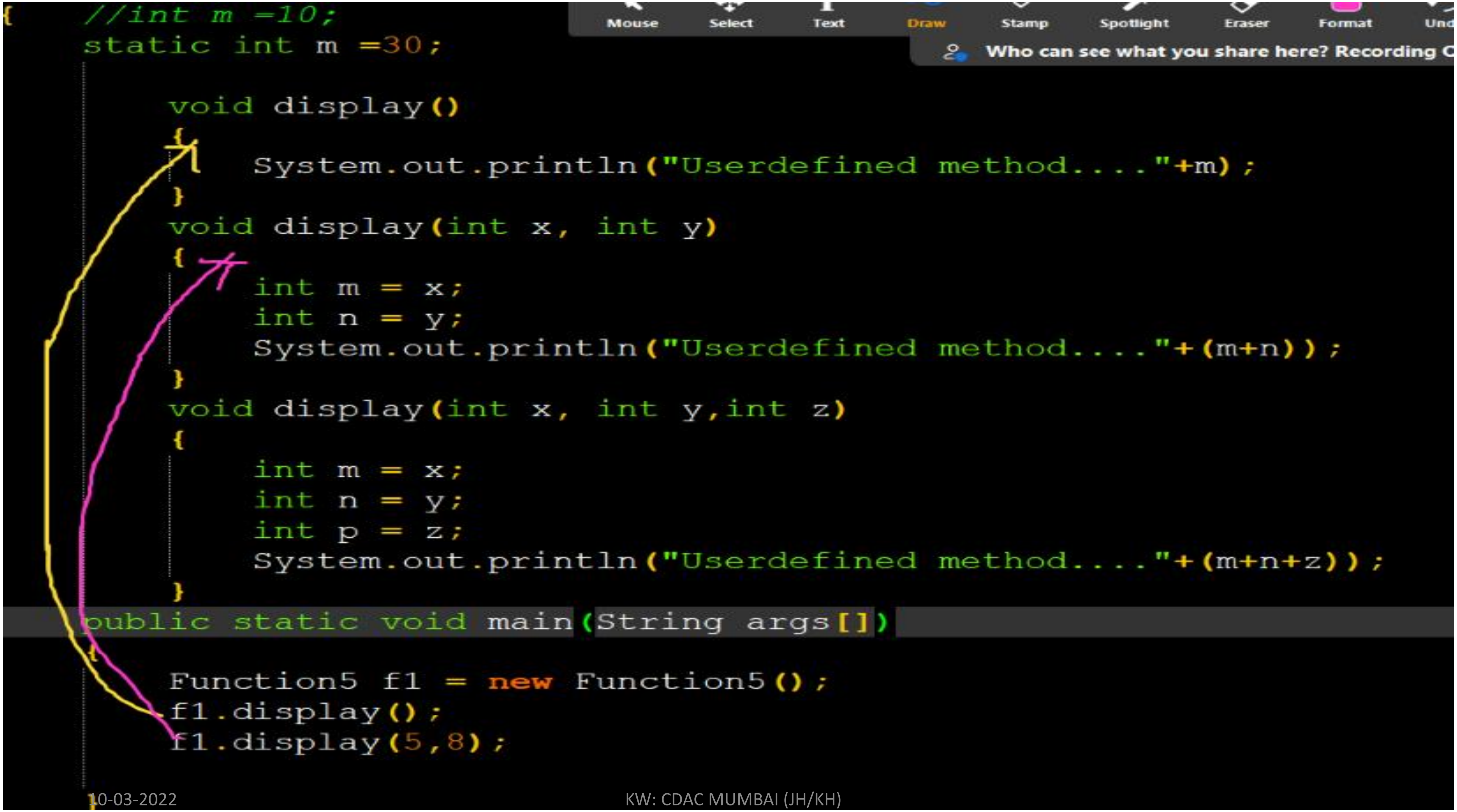
```
class Function2
{

    void display()
    {
        System.out.println("Userdefined method....");
    }
    void display1(int c)
    {
        int m = c;
        System.out.println("User
    }
  public static void main(String a
  {
    Function2 f1 = new Function2
    f1.display();
    f1.display1(10);

  }
}
```

Command Prompt

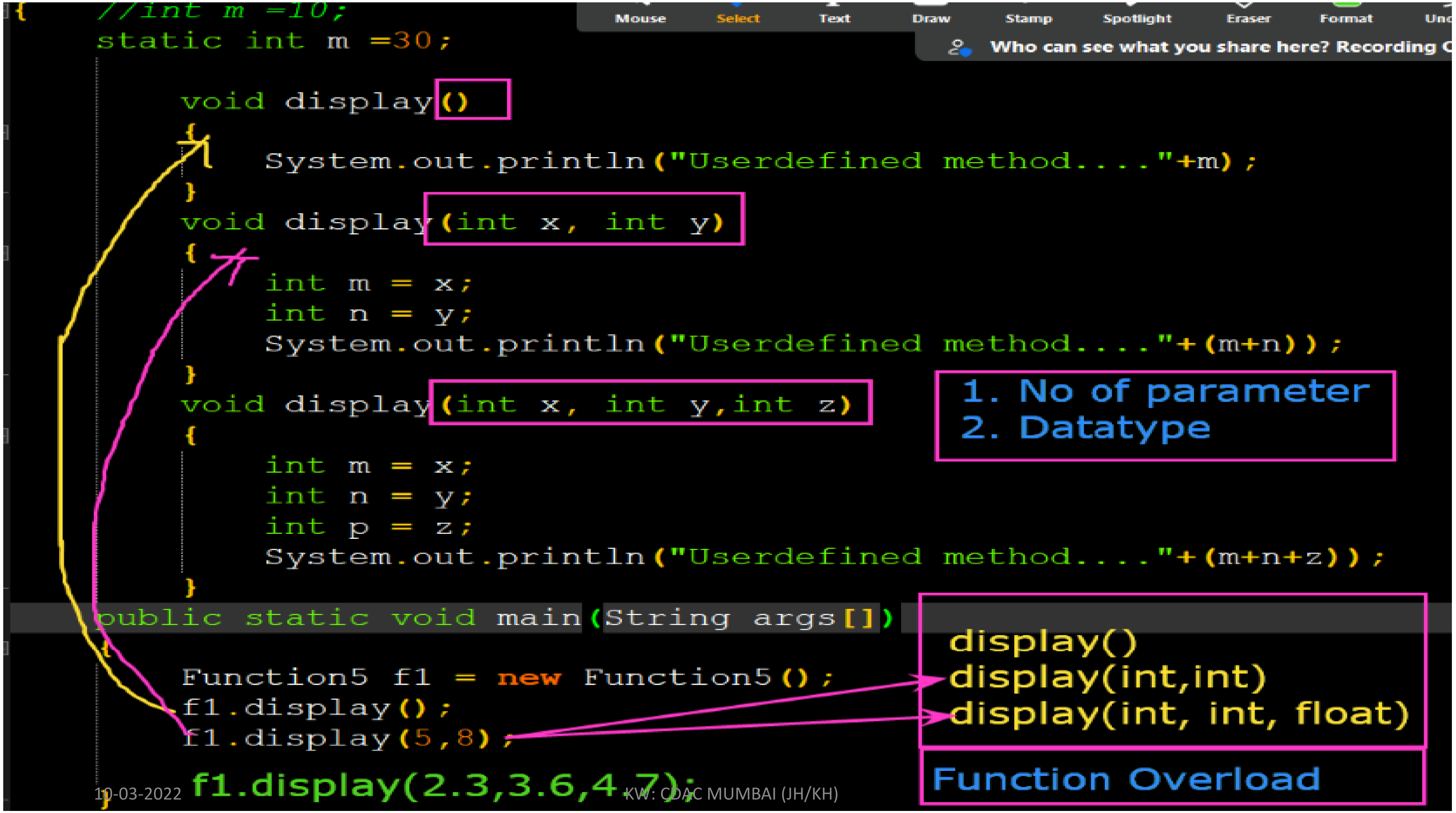
```
C:\CDAC22>javac Function2.java

C:\CDAC22>java Function2
Userdefined method....

C:\CDAC22>javac Function2.java

C:\CDAC22>java Function2
Userdefined method....
Userdefined method....10

C:\CDAC22>
```


Who can see what you share here? Recording On

```
class Function4
{   int m =10;

        void display()
        {
            System.out.println("Userdefined method....");
        }
        void display1()
        {
            int m =20;
            System.out.println("Userdefined method...."+m);
        }
    public static void main(String args[])
    {
        Function4 f1 = new Function4();
        f1.display();
        f1.display1();

    }
}
```



```
{     //int m =10;
      static int m =30;

          void display()
          {
                  System.out.println("Userdefined method...."+m);
          }
          void display(int x, int y)
          {
                  int m = x;
                  int n = y;
                  System.out.println("Userdefined method...."+(m+n));
          }
          void display(int x, int y,int z)
          {
                  int m = x;
                  int n = y;
                  int p = z;
                  System.out.println("Userdefined method...."+(m+n+z));
          }
      public static void main(String args[])
      {
          Function5 f1 = new Function5();
          f1.display();
          f1.display(5,8);
```

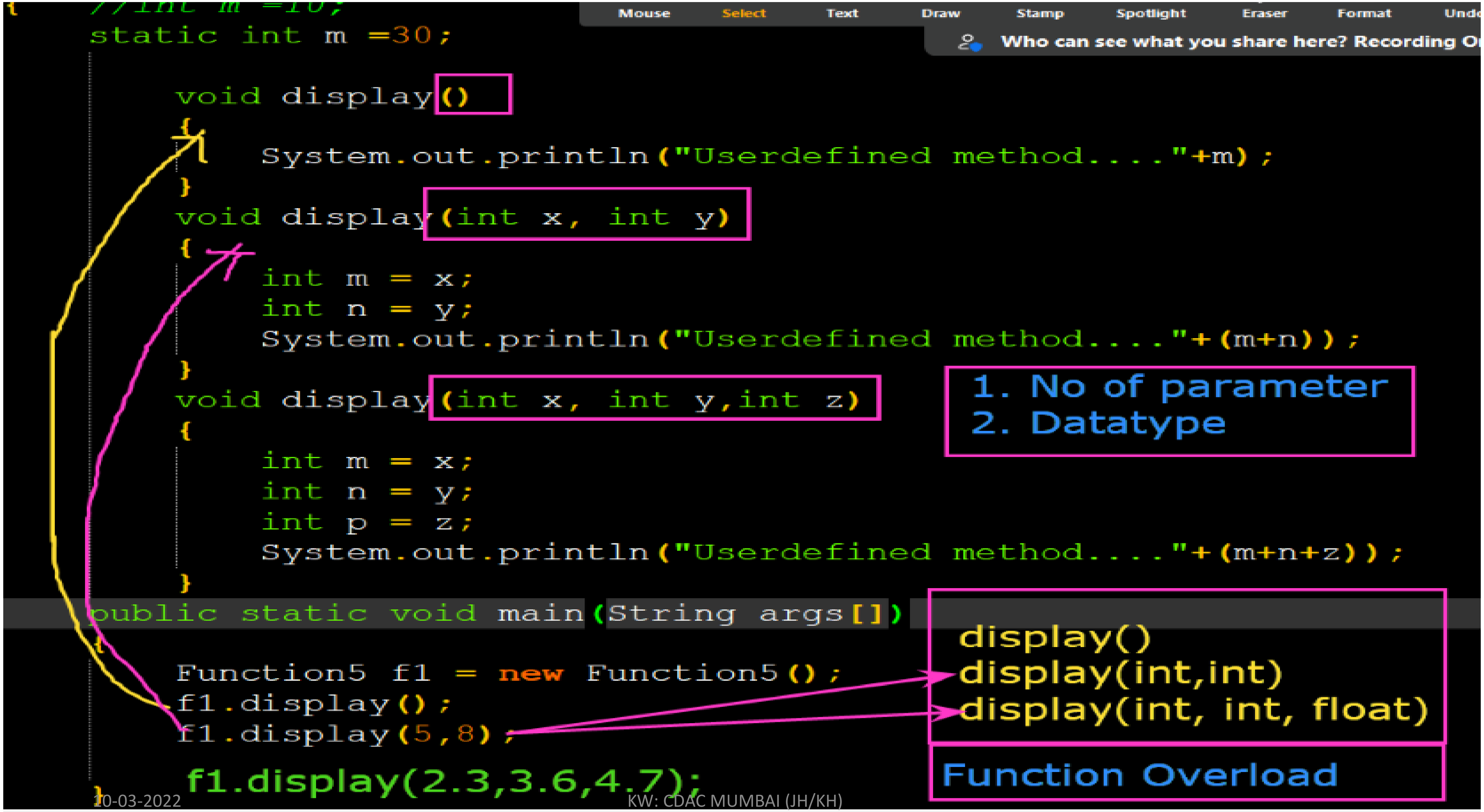


```
{       //int m =10;
        static int m =30;

            void display()
            {
                System.out.println("Userdefined method...."+m);
            }
            void display(int x, int y)
            {
                int m = x;
                int n = y;
                System.out.println("Userdefined method...."+(m+n));
            }
            void display(int x, int y,int z)
            {
                int m = x;
                int n = y;
                int p = z;
                System.out.println("Userdefined method...."+(m+n+z));
            }
        public static void main(String args[])
        {
            Function5 f1 = new Function5();
            f1.display();
            f1.display(5,8);
          f1.display(2.3,3.6,4.7);
}
```

1. No of parameter
2. Datatype

display()
display(int,int)
display(int, int, float)

Function Overload

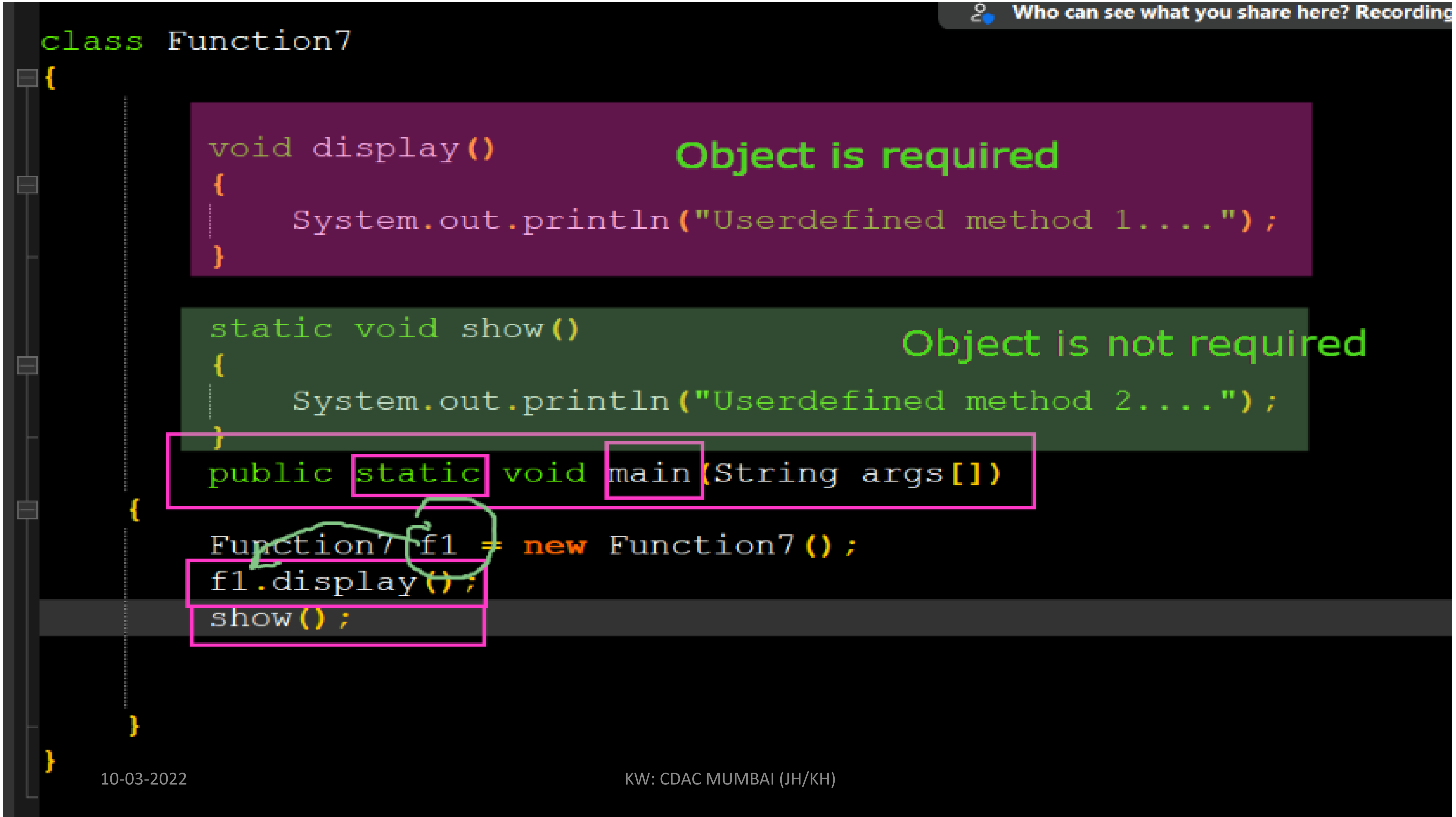


```
{       //int m =10;
        static int m =30;

            void display()
            {
                System.out.println("Userdefined method...."+m);
            }
            void display(int x, int y)
            {
                int m = x;
                int n = y;
                System.out.println("Userdefined method...."+(m+n));
            }
            void display(int x, int y,int z)
            {
                int m = x;
                int n = y;
                int p = z;
                System.out.println("Userdefined method...."+(m+n+z));
            }
        public static void main(String args[])
        {
            Function5 f1 = new Function5();
            f1.display();
            f1.display(5,8);
             f1.display(2.3,3.6,4.7);
        }
```

1. No of parameter
2. Datatype

display()
display(int,int)
display(int, int, float)

Function Overload



Who can see what you share here? Recording

```
class Function7
{
        void display()
        {
            System.out.println("Userdefined method 1....");
        }

        static void show()
        {
            System.out.println("Userdefined method 2....");
        }
        public static void main(String args[])
    {
        Function7 f1 = new Function7();
        f1.display();
        show();


    }
}
```

Object is required

Object is not required

10-03-2022