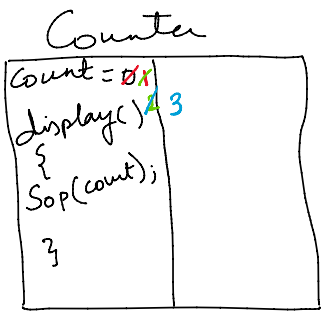
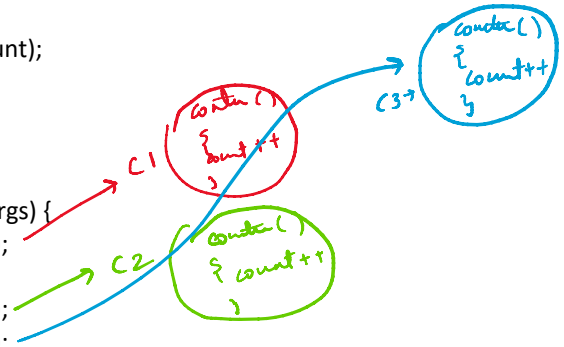
Stack and Object class

03 November 2022 07:31 PM

Use of static variable

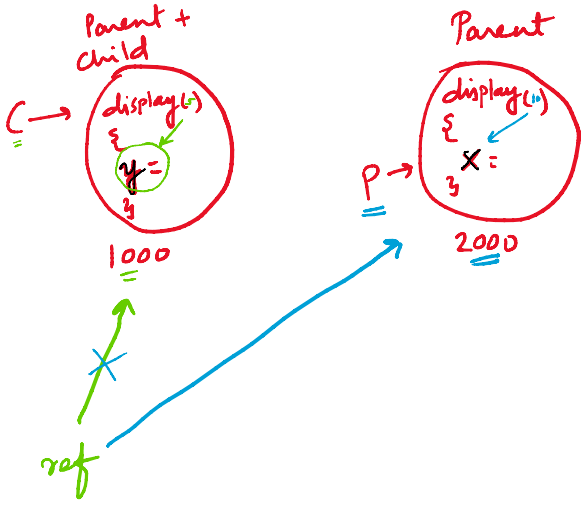
```
class Counter
{
   static int count;
   Counter()
   {
      count++;
   }
   static void display()
   {
      System.out.println("Count = "+count);
   }
}
public class Main
{
        public static void main(String[] args) {
              Counter c1=new Counter();
               Counter.display();
               Counter c2=new Counter();
               Counter c3=new Counter();
               Counter.display();
        }
}
```





Dynamic Method Dispatching

```
class Parent
{
   void display(int x)
   {
      System.out.println("X= "+x);
   }
}
class Child extends Parent
{
   void display(int y)
   {
      System.out.println("Y= "+y);
   }
}
public class Main
{
        public static void main(String[] args) {
               Child c=new Child();
               Parent p=new Parent();
               Parent ref;
               ref=c;
               ref.display(5);
               ref=p;
               ref.display(10);
        }
}
```



Private Variable

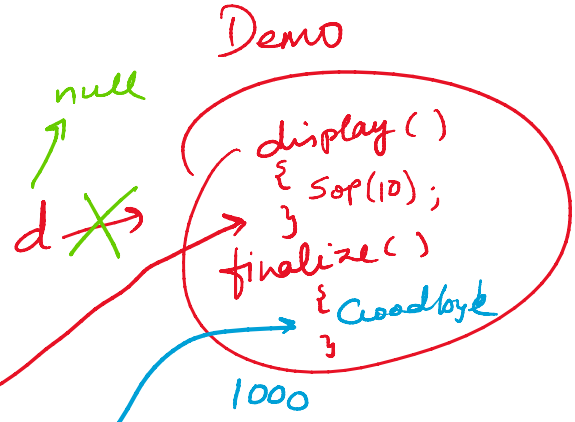
Private Variable.

```
class Test {
  private int age;
  public void setAge(int a)
  {
    if(a>18)
    age=18;
    else
    System.out.println("You are not allowed to vote");
  }
  public int getAge()
  {
    return this.age;
  }
}

class Main {
  public static void main(String[] args)
  {
    Test test = new Test();
    //test.setAge(4);
    //test.age=25;
    System.out.println("Age: " + test.getAge());
  }
}
```

finalize Method.

```
class Demo
{
   void display(int x)
   {
      System.out.println("X= "+x);
   }
   protected void finalize()
   {
      System.out.println("I m in the finalize method, good bye ");
   }
}
public class Main
{
      public static void main(String[] args) {
            Demo d=new Demo();
            d.display(10);
            d=null;
            System.gc();
      }
}
```



Stack class-

```
import java.util.Stack;
public class Main
{
public static void main(String[] args)
```

```
{
//creating an instance of Stack class
Stack<Integer> stk= new Stack<>();
// checking stack is empty or not
boolean result = stk.empty();
System.out.println("Is the stack empty? " + result);
// pushing elements into stack
stk.push(78);
stk.push(113);
stk.push(90);
stk.push(120);
//prints elements of the stack
System.out.println("Elements in Stack: " + stk);
result = stk.empty();
System.out.println("Is the stack empty? " + result);
}
}
```

Object class-

```
// Demo of Object class

class Demo
{
  void display(int x)
  {
    System.out.println("X= "+x);
  }
  protected void finalize()
  {
    System.out.println("I m in the finalize method, good bye ");
  }
}
public class Main
{
      public static void main(String[] args) {
            Demo d=new Demo();
            d.display(10);
            System.out.println("Object value is "+d);
            System.out.println("To string method of Object class is "+d.toString());
            System.out.println("Hash value is "+d.hashCode());
            Demo d1=new Demo();
            Demo d2=d1;
            System.out.println("Are two Objects equal?  "+d2.equals(d1));
            Object o=new Demo();
            System.out.println("o is an object of type  "+o.getClass());
            d=null;
            System.gc();
      }
}
```