What are the topics for tomorrow's session?
   Tommo topic: navin sir (SQL) for weekend batch enterprise java batch tommo at 9.00AMIST
                        JDBC(if time permits)

Sir cloneable small demo
Cloneable
   => it is a marker interface through which the objects can be cloned.

```
class Demo implements Cloneable
{
      int i;
      Demo(int i){
            this.i =i;
      }

}

Demo d1=new Demo(10);
Demo d2 =(Demo)d1.clone();
```

Q>
sir is interface which implemnets multiple inheritance?

```
   interface IDemo{ void m1();}
   interface ISample{void m2();}

   class SampleImpl implements IDemo,ISample{
            public void m1(){}
            public void m2(){}
   }
```

Q>
sir interface the word which works as srs and the interface which works as implementation of multipleinheritance is that same?
  Yes both are same..

Q>changes for mutable strings happen in existing object or a new object gets created &
     old reference points to a new object ? as you solved snippet with both concept, need clarity

```
Integer x = 400;//wrapper class immutable
Integer y = x; // y = 400
   x++; //x =401
System.out.println(x==y);//false

StringBuilder sb1 =new StringBuilder("sachin");//mutable
      sb1.append("tendulkar");//sb1,sb2 = sachintendulkar
StringBuilder sb2 =sb1;
System.out.println(sb1==sb2);//true
```

Q>Why can't we override a static method sir?
      static methods are dealing the information at class level,so overriding is not possible rather
      if we try to do it results in "Method hiding".

```
Q> interface A{}
class B implements A{}
      A a=new B();
```

Can we say refrence variable of A is the object of B?
 ans. true

Q>sir y cant we create class as static but inner class as satic
      Outermost class can't be marked as static becoz "static" refers to usage of class without creating an Object.
      where as inner class can be marked as static becoz "inner class" data can be refered without creating the outer class Object.

Q> Sir this is one question interview question. What does null mean in java.
            null in java represents a keyword where we can use it store the default value for reference variables.
                  eg: String name = null; Date d =null;

Q> foremost advantage of HAS A realationship. we can aslo build without realtionship aslo the output is same know sir?
            HAS-A => communication and navigation of data is made simple
            without HAS-A relationship the complexity of coding would increase.

Q> sir can you explain again why can't we override variables in interface?
      overriding is not applicable for variables,it is applicable only for methods

 Q> Can you give a brief about dependency injection?
      The process of injecting dependent object into target object is called "Dependancy injection".
            It can be done in 2 ways
                  a. constructor
                  b. setter

eg: Flipkart(object)//target object
         |
      DTDC(object)// dependent object

Q> what is tight coupling and loose coupling
            class Animal{}
            class Monkey extends Animal{}

            Monkey monkey = new Monkey();//tight coupling
            Animal animal   =Demo. getAnimal(); //loose coupling

class Demo{
      public static Animal getAnimal(){
            Animal animal =null
            //logic of creating object
            return animal;
      }
}

Q> what is override toString methos do??
       toString() -> it will be called automatically when we use object reference.
                     Normally in toString() we override to print the object data.

Q>
public enum Enum1 -> Enum1.class, Enum1$1.class, Enum1$2.class,Enum1$3.class
{
      RED(100)
        {
            public   void displaycolor()

```
		{
			System.out.println("your color name is "+this);
		}
	},

	GREEN(200)
	{
		public  void displaycolor()
		{
			System.out.println("your color name is ");
		}
	},

	BLUE(200)
	{
		public  void displaycolor()
		{
			System.out.println("your color name is ");
		}
	};

	private Enum1(int a)
	{
		System.out.println("I am in enum");
	}

	public  abstract  void displaycolor();

}

Ananomyous inner class
====================
final Parent  p =new Child(){

}
abstract class A{}
abstract class B extends A{}
abstract class C extends B{}
class D extends C{}
```