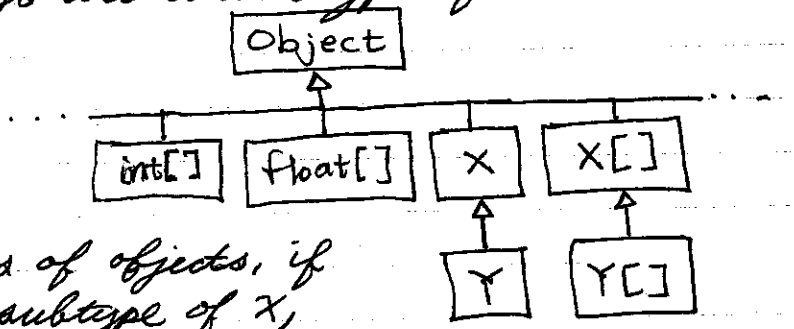
1. Explain how the subtyping relationship works for arrays in Java. Distinguish between the terms overloading and overriding.

All arrays are a subtype of the class Object:



* For arrays of objects, if Y is a subtype of X, then so is Y[] a subtype of X[]. *

Overloading is where you can create two in the same class methods of the same names, but they must both contain a different signature.

Ex: int add (int num1, int num2) add(int, int)
int add (float num1, float num2) add (float, float)

Overriding is where you, as a subclass, can provide a non-static method with the same signature, name, and return type of a method in the superclass. This new method replaces the one in the superclass.

* If it was a static method, then it would be hiding.

(2) Explain how name collision when multiple interfaces are implemented is handled in Java?

- If they have different signatures, they are considered overloaded
- If they have same signature and same return type, they are considered to be same method. In other words, the two methods collapse into one.
- If they have the same signature but different return types, a compilation error will result.
- If they have the same signature and return type but throw different exceptions, they are considered to be same method, and the resulting throws list is the union of the two original throws lines

#3 Let swap(arr, i, j) be a method which swaps the elements at index i and index j in array arr. Define what the term design by contract means and give a good contract for this method in the format we used in class for preconditions and postconditions.

Design by contract defines the conditions that are necessarily true at instant of method call and instant of method return.

```
@Pre   arr != null
@Pre   i >= 0 && j >= 0
@Pre   i < arr.length && j < arr.length
@Post  arr[i] == arr[j]@Pre
@Post  arr[j] == arr[i]@Pre
@Post  @forall k : [0..arr.length]@(k==i) ||
           (k==j) || (arr[k] == arr[k]@Pre)
```

#4

To ensure a java class is to be well behaved. A class must follow canonical form.

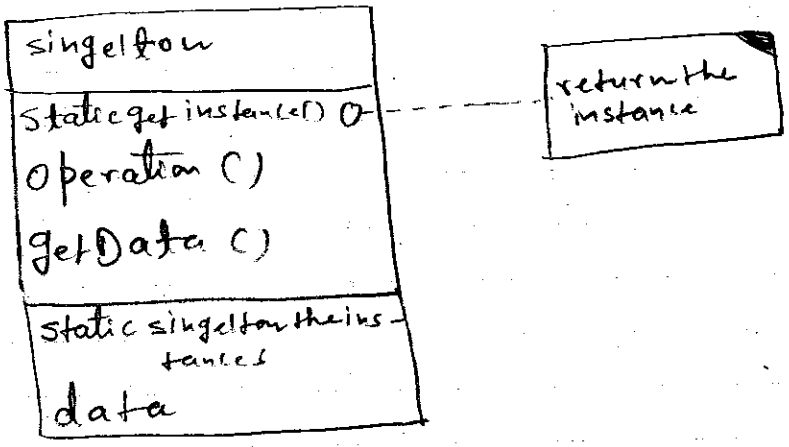
It means, a a java class provide public no arg constructor, an override method for hashCode(), equal(), toString(), an implementation of Cloneable interface, override the clone(), implement java.io.Serializable interface and override readObject() and writeObject() when it is needed to saved in files or transfer over the network. All the methods that are overriden must follow the specified contract for the interface.

#5 Singleton pattern

Category : Creational design Pattern

Intent : Make sure that class has only one instance.

Applicability: Use this pattern when a class must have only one instance and it must be accessible to clients.



Refactoring by delegation

```
Class A
{
  void stuff todo ()
    {
      helper . do stuff ();
    }
  Helper helper ;
}
```

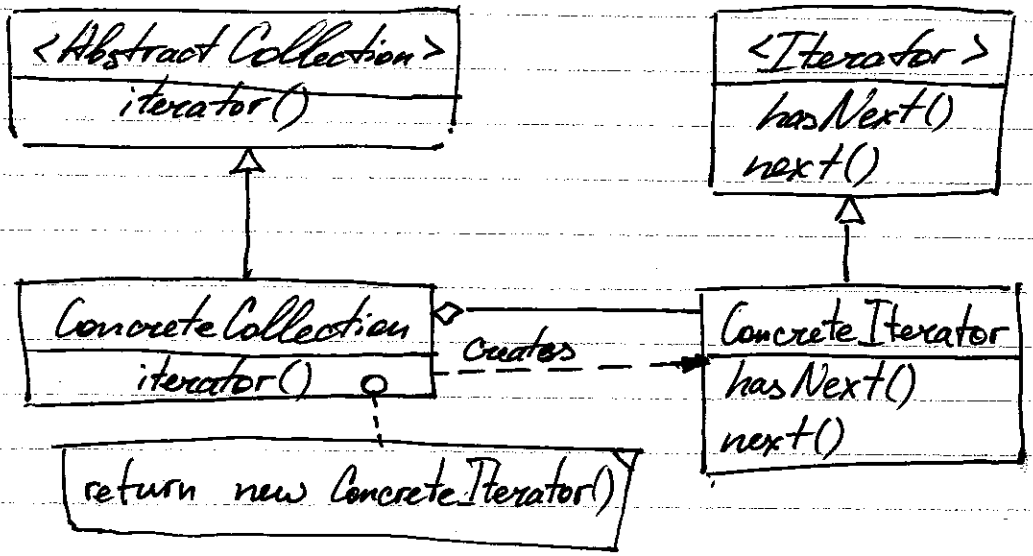
```
class helper
{ void do stuff ()
  { do one ;
    do two ;
  } do three ;
}
```

```
class B
{
  void stuff todo ()
    {
      helper . do stuff ();
    }
  Helper helper ;
}
```

Nikolai Efimov
SHAWN WyLLie
Suong Tran
CS 151

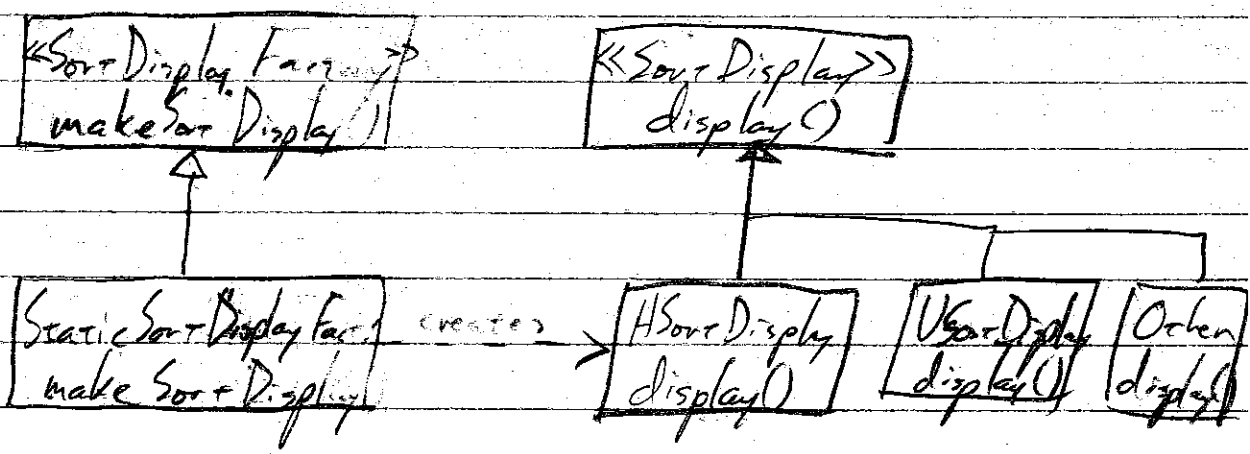
#6. a) Abstract coupling refers to how clients couple with service providers. A client only knows how to use a protocol and has no idea how service provider implements it.

b) Iterator design patter is a behavioral design pattern. It's used to provide a way to access elements of a collection sequentially.



c) ~~Be~~ Iterator Pattern is an example of an abstract coupling because we can have many different kinds of container classes and we can cycle through them without knowing how it's done inside of an implementation.

#7 Give an Example of the Factory design pattern.



The contract is enforced by the interfaces SortDisplayFactory & SortFactory, which decouples the Concrete object creation from the rest of the program.

StaticSortDisplayFactory is the «SortDisplayFactory» implementation that contains the actual switch() or Class.forName() stanza that creates the concrete SortDisplay-implementation objects.

#8) A Framework is a collection of classes, that represents reusable design.

For a Framework to be considered complete, it must have the following characteristics:

- Completeness
- Adaptability
- Efficiency
- Safety
- Simplicity
- Extensibility

Examples of Frameworks:

- Swing
- Collections
- A.W.T.

```
LinkedList<int> l;
l = new LinkedList<int>();
l.add(1);   <-
l.add(2);   <-  Autoboxing done automatically by Java 5.
l.add(3);   <-
```

~~foreach~~

```
for(int i : l)
    System.out.println(i);   <- Auto-Unboxing done automatically.
```

#9

```
public test extends JFrame implements ActionListener
{
   JPanel buttPanel;
   JButton buttReset;

   public test()
   {
      buttPanel = new JPanel();
      buttReset = new JButton("Reset");
      buttReset.addActionListener(this);
      buttPanel.add(buttReset);
      this.getContentPane().add(buttPanel, "South");
   }

   public void actionPerformed(ActionEvent e)
   {
      buttReset.setText("Hello");
   }
```

10 — not on test