1. Accept 2 numbers as command line arguments from user. If user supplies less t
han 2 arguments supply error message & terminate. . If all correct, compute avar
age & display the same.
2. Redo above assignement by replacing command line arguments by user inputs via
 scanner. If arguments are not numbers , supply error message & terminate.
3. Accept 2 numbers from user using scanner. Compare 2 numbers & print compariso
n results.
4. Display food menu to user. User will select items from menu along with the qu
antity. (eg 1. Dosa 2. Samosa .....10 . Generate Bill) When user enters 'Gene
rate Bill' option, display total bill & exit.

5.Consider the following. What will happen? MUST understand with memory diagrams
.

```
class Tank {
  private int level;
  Tank(int l)
   {
     level=l;
   }
  public void setLevel(int level)
  {
    this.level=level;
  }
  public int getLevel()
  {
    return level;
  }
}

public class Assignment {
  public static void main(String[] args) {
    Tank t1 = new Tank(10);
    Tank t2 = new Tank(20);

    s.o.p("1: t1.level: " + t1.getLevel() +
          ", t2.level: " + t2.getLevel());
    t1 = t2;
    s.o.p("2: t1.level: " + t1.level +
          ", t2.level: " + t2.level);
    t1.setLevel(27);
     s.o.p("3: t1.level: " + t1.level +
          ", t2.level: " + t2.level);
    t2.setLevel(t1.getLevel()+10);
     s.o.p("4: t1.level: " + t1.level +
          ", t2.level: " + t2.level);
  }
}
```

Day 2
1. Solve TestParamPassing4.java -- to understand passing & returning of array re
ference.
2.
Create Point class Point2D in package "geom" : for representing a point in x-y
co-ordinate system.
2.1 Create a parameterized constructor to accept x & y co-ords.
2.2Add show method : to return the x & y coords .(public String show())

2.3Add isEqual method : boolean returning method : must ret. true if both point
s are having same x,y co-ords or false otherwise. :
2.4 Add method in Point2D class , to return newly created point having given x
& y offset.
(eg Point2D createNewPoint(int xOffset,int yOffset))

2.4 Write TestPoint class with a main method --- in "tester" package.
Create 2 points by taking data from user. Use show method to display the co-ords
 of 2 points.
Use isEqual method : to chk the equality of 2 points.
Accept x offset & y offset from user & call createNewPoint method.
Display x,y co-ords of new point.

3. In TestBoxArrayScanner.java(from day1) --replace for loop by for-each, compi
le , run & understand differences.

Day 3
0. Explain how System.out.print or println(Object ref) --invokes ref.toString au
tomatically?
(Hint : search in javadocs of java.io.PrintStream)

1. Take earlier Point2D class & make following changes.

In Point2D class add --
1.1 double calcDistance(Point2D p) --which will return distance between current
point & supplied point
Distance formula
sqrt of (x2-x1) ^2 + (y2-y1)^2
Hint : Use java.lang.Math class methods --Math.sqrt & Math.pow for the computati
on.
1.2 Replace "show" method by overriding form of "toString" & "isEqual" by overri
ding form of "equals"
Must understand & use @Override annotation.

Write TestPoints class with a main method --- in "tester" package.
1.3. Prompt user for how many points to create & create suitable array for it.
1.4. Using for loop --prompt user for point co-ordinates & create Points.
1.5 Display distance between 1st & last point.
1.6 Using single for-each loop, display co-ordinates of all points in a loop(Use
 : toString)
1.7 Check if 1st & last point are equal (use : equals) & display "equal" or "not
 equal" according.

2. Create Utils class . Add a static method in Utils class to create and return
a new point accepting a point & x & y offset . Test it using Tester class's ma
in(..) method.

Day 4
1. Create Java application , for the following.
(Can use existing Emp,Mgr & worker classes or can create from scratch).
Tester class -- TestOrg --- tester pkg.
1. How many emps in org ?
2. Display options
2.1 Hire Mgr --- Accept mgr details or err mesg(Recruitment over) if array size
over.
2.2 Hire worker -- Accept worker details or err mesg if array size over.
2.3 Display Org details (emp info & net sal) -- single for-each loop

& display perf bonus if its mgr or rate if its worker
2.4 Fire Emp --- i/p id . Emp reference up-shifting is not expected.(only nullify the reference)
2.5 Display emp details -- i/p id & o/p --- emp not found OR actual; details
2.6 Update Salary of Emp --- i/p id , sal inc
2.7 Exit

2.

Create Java application for fixed stack & growable stack of employees functionality.(Hint : Stack interface & implementation class FixedStack & GrowableStack) From tester class --- display Menu
1 -- Choose Fixed Stack
2 -- Choose Growable Stack
Accept following options only after initial selection.
3 -- Push data
4 --- Pop data
5 -- Exit

Day 5 (optional : Please try after completing earlier assignments)

Create Java application for storing the customer details of the Online e-shop utility. Online part of e-shop will be later developed using the web-appln in advanced java.

Details about Customer :
(custId(int --auto increment) ,email(string : primary key),password(string)dateOfBirth(Date)

Add in Customer class -- equals method(email based)

Validations : 1. Customer's DOB : Between 1st Jan 1985- 31st Dec 1995
2. email must contain : @ character, min length 7 , max length =20
3. country code -- IN only
4. password : min length 5 , max length =10, should contain either -- * or $ or # to increase password strength.

In a Tester application : Create suitable array to store customers.

Support the following functionality via choices.

Choice 2. View All Customer details.(via for-each)

Choice 3 : Get Specific Customer Detail --- I/P Customer Email
O/p --- Via Custom exc -- Customer rec not found, pls register OR display Customer details.

Choice 4 : Un-subscribe Customer
I/P Customer email
O/p --- Customer rec not found, OR Customer un-subscribed message.

Choice 10 : Exit

Day 6
1. Must complete all earlier assignments.
2. Go through wrapper based ready made code samples & revise.

3. Go through Collection slides till slide 14 (at least)

Day 7
1.Compare Arrays.sort(Object[] arr) Vs Collections.sort(List<T> l1)
2. Replace array based customer assignment by ArrayList based implementation.
3. In earlier assignment --
3.1sort customers as per email (using Natural ordering)
3.1 sort customers as per date of birth (using Custom ordering)

Day 8
1. Clear all pending work & go through Map API + HashMap constructors.

Day 9.
1. Solve earlier Customer , ArrayList based assignment , by replacing ArrayList by HashMap or TreeMap suitably.

Remove compareTo , equals & email based constructor of Customer class.

Create Tester for the following
1. Register new customer
2. List all customers(via toString of HashMap)
3. Get Specific Customer Detail --- I/P Customer Email
O/p --- Via Custom exc -- Customer rec not found, pls register OR display Custom er details.
4. Un-subscribe Customer
I/P Customer email
O/p --- Customer rec not found, OR Customer un-subscribed message.
5. Update Customer --- update "password" of the customer.
I/(p -- email & old password.
If it matches -- replace old password by new password.
Otherwise throw suitable custom exceptions(Invalid customer email or invalid pas sword)
6. Display Customers sorted as per asc order of email
7. Display Customers sorted as per desc order of email
8. Display Customers sorted as per date of birth.

Day 10
Reading Homework only
1. MUST go through ready code samples of method overloading & understand the pre cedence(widening,boxing & var-args)
2. MUST read about "type erasure" from today's sequence file

3. Try to understand from java.util.Collections following methods(for clarity on generic syntax)
3.1 public static <T> List<T> nCopies(int n,
                                      T o)
3.2
public static <T> void fill(List<? super T> list,
                            T obj)
3.3
public static <T> void copy(List<? super T> dest,
                            List<? extends T> src)

Day 11.
1. Create Customer based utility(Customer class same) to
1.1 Restore customer details from text file , in buffered manner, at application start up.
1.2 Give options -- as Register Customer , Un subscribe customer , Sort customer s as per customer's date of birth,exit
1.3 Before terminating application --store customer details in text file (buffer

ed manner)