C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
1.variables defintion - data values are stored in memory location with specfic names.These memory locations are called variables
                        (think of shoe rack)
eg.int a=5,b=7;
int[] arr={};

2.keywords defintion -special words whose meaning is known to compiler
eg. public ,static,void,main
how many keywords are there in java 8->52

3.identifiers defintion - these are names given to differebt components of programs eg. class/interface,methods,variables
Rules for creating identifiers
    -does not starts with number
    -starts with letter,$ or _
    -no spaces allowed
    - hypen(-) not allowed
    -case sensitive
    - cannot be a keyword

4.Data Types are of 2 types
    -primitve : variable will hold value. eg.int,float,char
    -Non-primitve/Reference Data Tpes : variable will hold reference of memory location eg.String,User Defined data types

5.primtive data types
    -think of table(with memory size)
        boolean (1 bit depends on jvm)
        char (2 byte)
        byte(1 byte) short(2 byte) int(4 byte) long(8 byte)
        float(4 byte) double(8 byte)
    -Ascii values
        A(65) -Z(90)
        a(97) -z(122)
        0(48) -9(57)
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 1,561 Col : 8 Sel : 7 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 1:52 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPRHOE

```
6.operators
    -Arithemtic (+,-,*,/,%,++,--)
    -Relational (==,!='>,<,>=,<=)
    -Assigment (=,+=,-=,*=,/=,%=)
    -Logical (&&,||,!)
    -ternary operator (cond)?(statment if true):(statment if false)

7.Type Cobversion/Type casting (2 types)
    -narrowing type conversion(Explicit type casting) : large data type to small(think of bucket)
    -x(Automatic Tyoe casting) : small data type to large (think of bucket)

    eg a/b*c
        here a:short,b:long,c:float
        so a/b will give long .....long * float=float
        so answer is float
        value is decided in such a way that it can accomodate value equal to larger data type

8.Switch Statment: values can be byte,short,int,char,String
syntax:
    char var='A';
    switch(var){
        case 'A':System.out.println("case 1 executed");
                 break;
        case 'B':System.out.println("case 2 executed");
                 break;
        default:System.out.println("default");
    }

9.Array :is used to store similar type of data.--> Can be used to store data of primitive as well as reference data type.
         size is fixed at time of declaration.--> it is object in java and can be created dynamically.
         The length attribute of an array can be used to get its size
```

Windows (CR LF) UTF-8 INS

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROG

```
64          The length attribute of an array can be used to get its size
65
66          eg.
67          int myArray2[];
68          int []myArray5;
69          int[] arr1={1,2,3,4,5};
70          int[] arr2=new int[3];
71          int[] arr3=new int[]{1,2,3,4,5};
72          int[][] arr4={{1,2,3,4},{1,2,3},{1,2,3,4,4,5}};
73          int[][] arr5=new int[5][6];
74          int[][] arr5=new int[5][];//column number is optional
75          int[][] arr6=new int[][]{{1,2,3,4},{1,2,3},{1,2,3,4,4,5}};
76
77  # OOP CONCEPTS IN JAVA
78      Object
79      Class
80      Inheritance
81      Polymorphism
82      Abstraction
83      Encapsulation
84
85  10.class :is used to represent real time entity.
86           it has attributes(instance variables) and methods.
87           eg.Person is class
88                everyone around us is object eg.interviewer
89
90  public float someMethod(int price){
91  int totalPrice=price+50;
92  }
93
94  here --->price : formal Arguments or parameters
95          totalPrice ,price :Local variables
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 1,561 | Col : 8 | Sel : 7 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 1:53 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPRO

```
11.Object: is instance of class
eg.Product prodObj=new Product();
here prodObj is non-prmitive data type so it will store reference of memory location
prodObj is called reference variables.

Note:instance variables will be initialzed by their default values eg.int :0,String:null

12.Method Overloading :if a class have more than one method of same name
                       parameters differ on data type,no of parameters,order
                       call is resolved at "compile time".

13.Constructor :  is used for creating and initializing objects
                  is used to set values of instance variables/attributes at time of object creation.
                  A default constructor is already given by compiler which initialze attributes by default values.
                  constructor is special method whose name is same as class with no return type

    2 types :default constructor
             parameterized constructor

14.constructor overloading : more than one costructor differing by number ,type,order of arguments

15. Shadowing of a field: when local variable has same name as that of instance variable ,so local variable has more priority
eg.
class Demo{
   int id;
   String name;
   public void setId(int id){
       id=id;
   }
}

class StartMain{
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 1,561 Col : 8 Sel : 7 | 1 | Windows (CR LF) UTF-8 INS

Type here to search | 14°C Cloudy | 1:53 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPSQG

```
class StartMain{
    public static void main(String[] args){
        Demo ob=new Demo();
        ob.setId(5);
        System.out.println(ob.id);//output--->0(default value set on constructor initialization)
    }
}

16.this keyword : this solves problem mentioned in point 15
eg.
class Demo{
   int id;
   String name;



  public Demo(int id){
      System.out.println("inside constructor1");
      this.id=id;
  }

  public Demo(int id,String name){
      this(id);//calling one constructor into another constructor should be "1st statment"
      this();// in case if we have default constructor ....here we dont have
      this.name=name;
       System.out.println("inside constructor2");
  }
}

class StartMain{
    public static void main(String[] args){
        Demo ob=new Demo(3,"sdfsfds");
```

ormal text file | length : 53,304 lines : 1,609 | Ln : 1,561 Col : 8 Sel : 7 | 1 | Windows (CR LF) UTF-8 INS

Type here to search | 14°C Cloudy | 1:55 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPRING

```
157 class StartMain{
158     public static void main(String[] args){
159         Demo ob=new Demo(3,"sdfsfds");
160         //output--->inside constructor1
161                     inside constructor2
162     }
163 }
164
165 17.Abstraction and Encapsulation
166
167 Encapsulation:is binding of data and methods
168               methods are kept public while attributes are kept private so they are not directly manipulated (Why? see example)
169               eg. consider if we want to increase speed(attribute) of car(Class) .Also spped cannot be increased above 120km/hr
170                 so here speed if kept public can be varied above 120 also so it will be kept as private + method that control
                    speed
171                 will be having condition if speed cross >120 so we will keep as 120
172                 will be public
173                 Consider code :
174                 class Car {
175                         private float speed;                                    // private prevents direct access
176
177                         public void accelerate(float speedToAdd) {              // Method to increase speed
178                             this.speed += speedToAdd;
179                             if(this.speed > 120) this.speed = 120;              // Limiting speed to 120 kmph
180                         }
181                     }
182 Abstraction: telling what is does but not telling how it does
183              eg naming a method like getPriceAfterDiscount(15) so it defines what is does but does not define how it does
184
185 18.Static Keyword
186     Static can be used with variables,methods,static blocks
187     static variable,method,blocks are Class level property and does not belong to any object
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 1,561 | Col : 8 | Sel : 7 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 1:56 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROG

```
18.Static Keyword
    Static can be used with variables,methods,static blocks
    static variable,method,blocks are Class level property and does not belong to any object
    static variable and blocks are initialzed on load of class
    static blocks are required to "initialze static variable" when cannot be done in one line (if manipulation required to
    initialze variable)

    STATIC VARIABLE: is used to generate id
        eg : class Product{
        private int id;
            private static int counter=1000;
            public Product(){
                id=counter++;//evertime a new object is created so id will take new value
            }
        }

    STATIC METHOD : it can be called using class name or with reference/object
    eg. getProductId is static method so it can be called
            -Product.getProductId();
            -Product p1=new Product();
             p1.getProductId();

    STATIC BLOCK: it gets executed once when class is loaded .if multiple static blocks are there they are executed in order
    as written

STATIC CONTEXT : static blocks and methods can access only static members .it cannot access non static members as they belong
to class
                    not to any object
                    eg.public static void generateProductId() {
                            this.productId = ++idCounter;                    // Which object's productId!?(COMPILATION
                            ERROR!!!)
                            }
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 1,561 Col : 8 Sel : 7 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 1:56 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SRPROG

```
        NOTE: non static members can access static and non static members

19.Inheritance : when a class inherits attributes and methods of another class
                              Or
                 when child class inherits properties of parent class
                 "is-a" relationship
                 eg.smart Tv is a Tv , Mobile is a product

                 Code example :
                 class Customer {
                        // Parent/Super/Base class
                    }

                    class RegularCustomer extends Customer {        // RegularCustomer is a Customer
                        // Child/Sub/Derived class
                    }

                    class PremiumCustomer extends Customer {        // PremiumCustomer is a Customer
                        // Child/Sub/Derived class
                    }

Super Constructor: when child class constructor(default/parameterized) is called ,it will automatically call default
super/base class constructor.
                   we do not have to mention super() for default constructor if we want to we can write.
                   If base class constructor is parameterized so call from child class constructor has to be done otherwise it
                   will throw
                   compilation error.ALso call to parent constructor should be 1st statment

Types of inheritance: single
                      hierachial (more than one child class has same base class)
                      multilevel
(Not supported in java)multiple (more than one parent class with one child class)-Can be achieved through interfaces
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 1,561 | Col : 8 | Sel : 7 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 1:57 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROG

```
243
244 "Super keyword is used for parent class"
245
246 NOTE: if in derived class we want to call parent class method or non-private instance variable then we can use
    super.parentClassMethod() or
247 super.variableName in childClassMethod() or constructor
248
249 20.Aggregation : when an object contains another object ,it is called aggregation.
250                  it is "has-a" relationship
251                  it is given as instance variable
252                  eg.customer has a address
253
254 21.Association : when we use object of one class in parameters or local variable of methods of another class
255                "uses-a" relationship
256
257 22.Method Overriding : when parent class is inherted by child class then child class can "redefine" method with its own
    implementation.This
258                       is called method overriding
259                  NOTE-method signature should be same as that of parent class
260                      -"overriding method should not have weak access specifier than parent class"
261                      -private members not inherted hence not overriden
262
263 Call to a particular method is decided at Run time so called "Dynamic Binding"
264 Call to a particular method is decided at Compile time so called "Static Binding"
265
266 Code Example1:
267         Customer customer = new RegularCustomer();
268         customer.orderProducts(products);
269
270 Code Example2:*****
271         public static void showCustomerDetails(Customer customer) {
272             customer.displayDetails();
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 1,561 Col : 8 Sel : 7 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 1:57 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROG

```
Code Example2:*****
        public static void showCustomerDetails(Customer customer) {
            customer.displayDetails();
        }

        showCustomerDetails(new RegularCustomer(101, "Tony", "tony123", "9876543210", new Address("1 Hebbal", "Mysore",
        570027));
        showCustomerDetails(new PremiumCustomer(102, "Banner", "banner123", "9876543211", new Address("2 Hebbal", "Mysore",
        570027));

NOTE:only inherted and overriden methods are called using parent class reference.Any new method created in child class will
not be accessible
      static methods are inherited
      static methods are not overriden(@override use krenge to compilation error dega)
      final methods cannot be overriden

what kind of method cannot be inherited ----->private
what kind of methods can be inherited but not overriden --->static and final methods

lets Summarize:*****
classA----inherits-----classB
classA obA=new ClassA();---->all methods and variables of classA accessible
ClassB obB=new ClassB();----->all methods and variables of classB +inhertied from classA accessible
classA obA=new ClassB();----->all methods and variables of class A inherited +overriden methods from class A to B can be
accessed

23.Final keyword :can be used with variable,method,class
                Final variable -final field value cannot be changed once it is initialized.
                               eg private final float Pi_value=3.14f;
                Final Method - final method cannot be "overriden" in sub class.
                Final class - final class cannot be extended
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 1:57 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
24.Abstract keyword :can be used with class and method
                     abstract means something is missing (adhura)
                     if we want to make sure that child class must implement a method then that method should be made abstract
                     in
                     parent class....if method is made abstract ..now class is also needs to be made abstract otherwise class
                     would be incomplete

Abstract class is like any other class where we can declare variables,declare abstract methods,define non abstract methods
everything
but "it cannot be instantiated ....it is used for inheritance"
NOTE: -abstract method cannot be private ,final and cannot have definition in abstract class
      -abstract is used when we dont have complete class/method
      -when we dont want class to be instantiated.
      eg. AbstractClass ac=new AbstractClass();//Compilation error

-----------------------------------------------
Abstract classes enforce inheritance          |
and                                           |
Abstract methods enforce overriding           |
And we get another flavour of dynamic binding |
-----------------------------------------------
if child class does not override all abstract methods of parent class thenchild class is also made abstract


Simple/Concrete class                |       Abstract class
-------------------------------------------------------------------
instantiated                         |       not-instantiated
cannot have abstract method          |       can have
no force to overide methods          |       forcefully have to overide method(wat needs to be done is known but how it is done
based on child class)

25.Interface:it is "generic template" which shows all variables and methods are "declared"
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 1:58 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt

```
25.Interface:it is "generic template" which shows all variables and methods are "declared"
             (wat needs to be done is known but how it is done based on child class)
              variables are initialized in interface
                 eg.public interface Card {
                     int cardid;//COMPILATION ERROR
                 }
                 CORRECTION:
                 public interface Card {
                     int cardid=10;
                 }
abstract class can do same what we can do with inteface but multiple inheritance is key of suceess(exclusive property) for
inteface

Important Points:In java 8 interface has static and default methods
                 methods declared are public and abstract
                 data fields declared public,static and final
                 class extend from same class and at same time implement any number of interface
                 class must override all methods from interface otherwise it will be made abstract

static and default method in interface
Code Example :
            public interface Card {
                 public default  void hello1() {
                     System.out.println("hello1");
                 }

                 public static  void hello2() {
                     System.out.println("hello2");
                 }

                 public void hello3();
            }
```

text file length : 53,304 lines : 1,609 Ln : 275 Col : 1 Sel : 2 | 1 Windows (CR LF) UTF-8

Type here to search 14°C Cloudy 1:58 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
343 static and default method in interface
344 Code Example :
345             public interface Card {
346                     public default  void hello1() {
347                         System.out.println("hello1");
348                     }
349
350                     public static  void hello2() {
351                         System.out.println("hello2");
352                     }
353
354                     public void hello3();
355             }
356
357             public class Customer implements Card{
358
359                 @Override
360                 public void hello3() {
361                     System.out.println("dfjhsdagjf");
362                 }
363             }
364
365             public class StartMain {
366                 public static void main(String[] args) {
367                     Card.hello2();//static method directly handled
368                     WRONG--xxx--Customer.hello2();----xxx--WRONG //Compilation error
369                     Customer c=new Customer();
370                     c.hello1();
371                     c.hello3();
372                     //c.hello2();//why it is not coming
373                     //is it inherited ? it should be
374
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 INS

Type here to search | 14°C Cloudy | 1:59 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SRPROC

```
78
79  focus1:static method can be called using interfacename and dot(.) operator
80  focus2:static method cannot be called using child class name/reference of child class name and . operator even though it is
    inherited
81
82   Abstract class                       |      Interface
83  ------------------------------------------------------------------------
84  cannot instantiated                   |       cannot instantiated
85  can have abstract method              |       can have
86  instance variable auto initialized    |       instance variable have to be initialized bcoz it is public static final
87  simple methods +abstract methods      |       static +default +abstract methods
88  soes not support multiple inheritance |       support multiple inheritance
89
90  Q.Why it is required to have default and static method in interface ?
91  Ans.bcoz suppose we need to add one common method(functionality) to all classes that implement interface so that can be done
92      static methods can be called using interface name
93      default methods are written in interface so that every class implenting interface need not to override it.we can override
        it if needed
94
95  if class implements multiple interface having same default method name ,it has to override them .Also "default" method of
    interface can be called using
96  Interface1.super.method_name() /  Interface2.super.method_name()
97
98  Syntax level problem clear:
99  ClassA extends ClassB
100 InterfaceA extends InterfaceB
101 ClassA implements InterfaceA
102
103  26.Object Class: it is super class of all classes.
104                  it provides some useful methods such as
105                     -equals()    : by default it uses memory address of object to compare for equality(just like ==)
106                                  : public boolean equals(Object obj) {...}
```

ormal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 1:59 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SEPROC

```
402
403 26.Object Class: it is super class of all classes.
404                 it provides some useful methods such as
405                    -equals()    : by default it uses memory address of object to compare for equality(just like ==)
406                                 : public boolean equals(Object obj) {...}
407                    -hashcode()  :it produces hash value of 32 bit integer bases on memory address
408                                 :public int hashCode() {...}
409                    -toString()  :it represent textual reprsesntation of object
410                                 :public String toString() {...}
411                                 eg. emart.Customer@af7d0676(fully qualified class name +@+unsigned hexadecimal reprsesntation
                                    of hashcode)
412
413
414 scenario of why equals and hashcode is used:if we want to avoid dublication of object then we override equals() and hashcode()
415 NOTE :if two object are equals according to equals() method then hashcode() should produce same value for both
416      :also if for two object hashcodes are equals it does not mean that objects are equal
417
418 Code Example: if id and name are equal then objects are equal
419 public class Product {
420     int id;
421     String name;
422     public Product(int id,String name) {
423         this.id=id;
424         this.name=name;
425     }
426     @Override
427     public int hashCode() {//No change in this
        method
428         final int prime = 31;
429         int result = 1;
430         result = prime * result + id;
431         result = prime * result + ((name == null) ? 0 : name.hashCode());
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 2:00 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 35 | new 37 | new 38 | SPRING

```
427     public int hashCode() {//No change in this
        method
428         final int prime = 31;
429         int result = 1;
430         result = prime * result + id;
431         result = prime * result + ((name == null) ? 0 : name.hashCode());
432         return result;
433     }
434     @Override
435     public boolean equals(Object obj) {//here are code changes
436         Product other=(Product) obj;
437         if(this.id==other.id && this.name.equals(other.name)) {
438             return true;
439         }
440         return false;
441     }
442 }
443
444 public class StartMain {
445     public static void main(String[] args) {
446         Product p1=new Product(1, "pawan");
447         Product p2=new Product(1, "pawan");
448         System.out.println(p1);              Output :  Demo1.Product@762efe5d  |  Demo1.Product@6582f33
449         System.out.println(p2);              Output :  Demo1.Product@71dac704  |  Demo1.Product@6582f33
450         System.out.println(p1==p2);          Output :  false                   |  false
451         System.out.println(p1.equals(p2));   Output :  false                   |  true
452         System.out.println(p1.hashCode());   Output :  1982791261              |  106442547
453         System.out.println(p2.hashCode());   Output :  1910163204              |  106442547
454     }
455 }
456
457 observations:1-memory address is also same for both p1 and p2 after overriding of equal() and hashCode().If we override only
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 2:01 PM 1/25/2022

```
456
457 observations:1-memory address is also same for both p1 and p2 after overriding of equal() and hashCode().If we override only
    equals and
458                 not hashcode then memory address would not have been same
459             2- p1==p2 is false in all cases
460
461 NOTE:The String and wrapper classes already override equals() and hashCode().
462
463 27.Comparable Interface has compareTo() function which can be used not only to check equality of objects but also for ordering
    of objects
464             in case objects are not equal
465 when overriden it should return :
466     -negative value when object is less than specified object
467     -zero if objects are equals
468     -positive value if objects is more than specified object
469
470 Code Example :
471 // Implementation of Comparable interface by overriding compareTo()
472 class Customer implements Comparable {
473     private int customerId;
474     private String name;
475     private String phone;
476
477     public Customer(int id, String name, String phone) {
478         this.customerId = id;
479         this.name = name;
480         this.phone = phone;
481     }
482
483     // Compares the current object with  the specified object
484     // Returns integer value according to the implementation
485     // Helps in sorting objects in a specific order
```

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
477     public Customer(int id, String name, String phone) {
478         this.customerId = id;
479         this.name = name;
480         this.phone = phone;
481     }
482
483     // Compares the current object with  the specified object
484     // Returns integer value according to the implementation
485     // Helps in sorting objects in a specific order
486     public int compareTo(Object o) {
487         Customer other = (Customer) o;
488         return this.customerId - other.customerId;
489     }
490 }
491
492 class ComparableDemo {
493     public static void main(String[] args) {
494         Customer c1 = new Customer(105, "Max", "8574637281");
495         Customer c2 = new Customer(109, "Nick", "9453281756");
496         if(c1.compareTo(c2) < 0) System.out.println("c1 comes before c2");
497         else if(c1.compareTo(c2) > 0) System.out.println("c1 comes after c2");
498         else System.out.println("c1 and c2 are same");
499     }
500 }
501
502 28.Important example :
503 public static void show(Object obj) {
504     System.out.println(obj);
505 }
506
507 here int ,boolean can also be passed bcoz primitive data types will be autoboxed  to respective wrapper class
508
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 2:02 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
29.String : Array of characters
            java.lang package
           immutable :cannot be changed after initialization
            Code example:
            String  a="pawan";
            for (int i=0;i<a.length();i++) {
                System.out.println(a.charAt(i));
            }
    String Methods
     CharAt(index)--return char
     indexOf(string/char)----return int
     contains(string)---return true/false
     substring(initial index,final index+1)
     split("anything")----return array of strin
     replace("with what","to what")----return void
-------------------------------------------------------------------------------------------------------------------------
---------------
30.Collection Framework:
It contains classes and interface to store and manipulate group of objects.

we will focus on:
-Collection data(List,Set,Queue)
-Map Data
-Cursors(To read collection data)
-Sorting (comparable and comparator interface)
-Stream Api(filter,sort and many more...)

Q.why we need collections bcoz we already have array?
   Array                              vs                    Collection
-can store primitve and non primitive data      -non-primitive(Object) data
-fixed size                                     -automatic size can be increased or decreased
-no methods(operation become complex)           -methods(operations become easy)
```

ormal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 2:03 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
Q.why we need collections bcoz we already have array?
   Array                                vs                    Collection
-can store primitve and non primitive data       -non-primitive(Object) data
-fixed size                                      -automatic size can be increased or decreased
-no methods(operation become complex)            -methods(operations become easy)
-performace Good                                 -performce is low bcoz whenever size is increase (copy old data into new
collection created eg. array list)

Collection f/w (java.util pkg)                                  vs Collections
root interface of Collection framework       it is class contains methods :sort() nCopies() reverse()


collection f/w divided into 2 parts:
                |
    -------------------------
    |                       |
a)Collection                b)Map
    -List
    -Set
    -Queue


                                                Itearble(I)
                                                Collection(I)
    List(I)                                     queue(I)                                        set(I)
ArrayList  LinkedList   Vector                  Deque(I)            PriorityQueue       SortedSet(I)        Hashset
                        Stack                   ArrayDeque                              TreeSet             LinkedHashSet


the root interface of Collection f/w is :Collection Interface
the root interface of Map is :Map Interface
collection frame is nothing but java.util pkg
the parent interface of collection is :Iterable Interface   (public interface Collection<E> extends iterable<E>{...})
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 2:04 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
566
567  -------Key Interface of Collection f/w----
568       Collection Data
569  Collection
570  List
571  Set
572      SortedSet
573      NavigableSet
574  Queue
575      Deque
576
577         Map Data
578  Map
579  SortedMap
580  NavigableMap
581  Map.Entry
582
583         cursors
584  Enumeration
585  iterator,
586  ListIterator
587
588         Sorting
589  Comparable
590  comparator
591  -------------------------
592
593              List Interface
594            Itearble(I)
595
596            Collection(I)
597
```

ormal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 2:04 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
593             List Interface
594            Itearble(I)
595
596            Collection(I)
597
598            List(I)
599  ArrayList(c)   LinkedList(c)              Vector(c)   1.0
600      1.2              1.2                  Stack(c)       1.0
601
602  Collection f/w introduced in java 1.2
603  Became Popular in java 1.5 : bcoz of AutoBoxing and Generics
604
605  classes introduced in java 1.0 version are called : Legacy Classes
606
607  Q.every collection class contains group of object as single entity then we so many classes like ArrayList,LinkedList,etc
608  Ans : every class have some features/characteristics
609
610  Characteristics of Collection f/w classes:
611  1.null insertion allowed
612  2.duplicate obj :allowed or not
613  3.synchronized or not synchronized   : java 1.0 sync rest all are non sync
614     t1              t1,t2,t3
615  4.insertion order
616        insertion     output
617        e1,e2,e3      e1,e2,e3 ------insertion order preserved (eg. all list classes)
618        e1,e2,e3      e3,e1,e2 ------ not preserved (eg. in set classes for some insetion order preserved and for some its not)
619  5.every collection class have underlined data structure (eg.for ArrayList it is array,for DoublelInkedList it is linked list)
620  6.version
621  7.heterogenous data can be stored except
622      TreeSet     TreeMap   -->both store homogenous data
623  8.cursors
624
```

ormal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 2:04 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 35 | new 37 | new 38 | SPPROC

```
626 ArrayList Characteristics:
627 1.null allowed
628 2.duplicate allowed
629 3.non sync
630 4.insertion order preserved
631 5.underlined ds is array
632 6.version 1.2
633 7.heterogenous
634 8.cursors
635
636 Code Example:
637 eg.List list=new ArrayList();//focus on declaration
638 list.add(10);
639 list.add(10.5);
640 list.add('m');
641 list.add(true);
642 we already discussed that we cannot add primitive data to collection but here we are adding in above exaple how ? so answer is
    AUTOBOXING
643
644 AutoBoxing:automatic conversion of primitive to wrapper object
645 int ---Integer
646 double---Double
647 char----Character
648
649 Arraylist,String,StringBuffer,wrapper objects has overriden toString() so whenever we print object it will show data
650
651 ***** autoboxing magic thats why collection become popular ****
652 before java 1.5 :
653 List list=new ArrayList();
654 list.add(Integer.valueOf(10));
655 list.add(Double.valueOf(10.5));
656 list.add(Character.valueOf('m'));
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 2:05 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
651 ***** autoboxing magic thats why collection become popular ****
652 before java 1.5 :
653 List list=new ArrayList();
654 list.add(Integer.valueOf(10));
655 list.add(Double.valueOf(10.5));
656 list.add(Character.valueOf('m'));
657 list.add(Bollean.valueOf(true));
658
659 AFter java 1.5 :
660 List list=new ArrayList();
661 list.add(10);
662 list.add(10.5);
663 list.add('m');
664 list.add(true);
665
666
667 ****Generics magic****
668 Arrays are type safe
669 but collections are not type safe so problems occurs are
670         ,       -Type Checking
671                 -Type Casting
672
673 eg.List list=new ArrayList();
674 list.add(new Student(101,"Pawan"));
675 list.add(new Employee(1001,"John"));
676
677 for(Object o:list){
678         if(o instanceOf Student){ ------->Type Checking
679         Student s=(Student) o;---------->Type Casting
680         sysout(s.id+" "+s.name)
681         }
682 else if(o instanceOf Employee)){...}
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 2:05 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

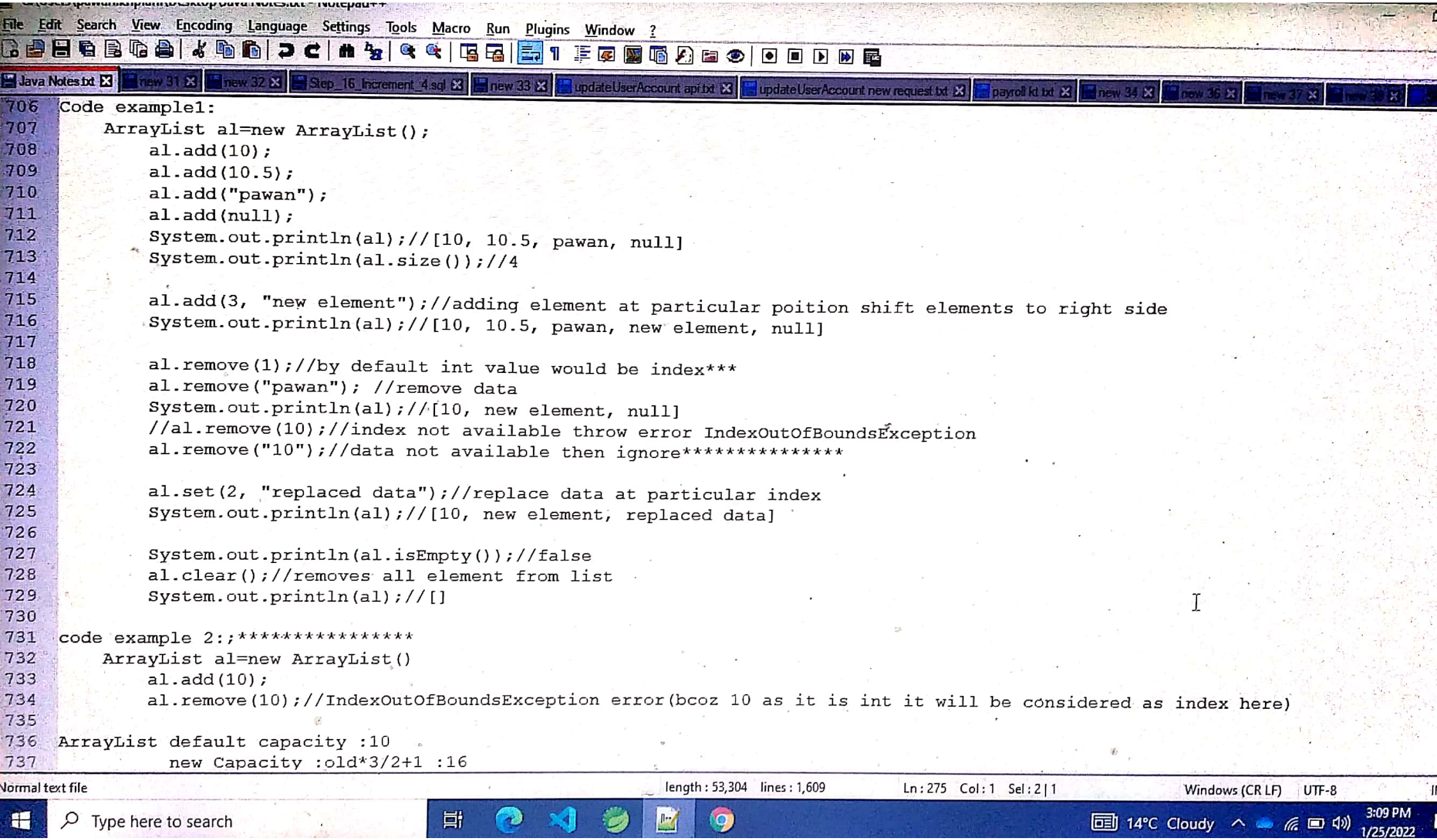
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
else if(o instanceOf Employee)){...}
}

So due to Problem of  Type Checking and Type Casting --->generics is introduced
Code example :Observe declaration

List<String> al=new ArrayList();
        al.add("pawan");
        al.add(1);-------> compilation error

List al=new ArrayList<String>();
        al.add("pawan");
        al.add(1);

ArrayList Methods:
    add(element)
    add(position,element)
    remove(index)
    remove(element)
    set(index,newElement)---->replace
    clear()
    size()
    isEmpty()

Code example1:
    ArrayList al=new ArrayList();
        al.add(10);
        al.add(10.5);
        al.add("pawan");
        al.add(null);
        System.out.println(al);//[10, 10.5, pawan, null]
        System.out.println(al.size());//4
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:09 PM 1/25/2022



C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROG

```
ArrayList default capacity :10
          new Capacity :old*3/2+1 :16
Arraylist Construtors
    -new ArrayList();
            default capacity :10
            new Capacity :old*3/2+1 :16
    -new ArrayList(int capacity);
    -new ArrayList(Collection c);
          adding one collection to another

Adding one list to another:
    approach1:using constructor
        List<Integer> l1=new ArrayList();
        l1.add(100);
        List<Integer> l2=new ArrayList(l1);

    approach2:using method
    List<Integer> l1=new ArrayList();
        l1.add(100);
        List<Integer> l2=new ArrayList();
        l2.addAll(l1);

Code Example : add(),addAll()
               contains(),containsAll()
               remove(),removeAll()

        List<String> list1=new ArrayList<>();
        list1.add("String1");
        list1.add("String2");
        List<String> list2=new ArrayList<>();
        list2.addAll(list1);
        list2.add("String3");
```

ormal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:10 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
776 Code Example :retainAll()
777         List<String> list1=new ArrayList<>();
778         list1.add("String1");
779         list1.add("String2");
780         List<String> list2=new ArrayList<>();
781         list2.addAll(list1);
782         list2.add("String3");
783         list2.add("String4");
784         list2.retainAll(list1);
785         System.out.println(list2);//[String1, String2]
786
787 Summarize all ArrayList functions:
788                 add(element)
789                 add(position,element)
790                 addAll(list)
791                 remove(index)
792                 remove(element)
793                 removeAll(list)
794                 contains(element)
795                 containsAll(list)
796                 retainAll(list)
797                 set(index,element)
798                 clear()
799                 size()
800                 isEmpty()
801
802 Code Example: Converting Array to List
803         String[] a1= new String[]{"abc","def","ghi");
804         List<String> l1=new ArrayList<>(Arrays.asList(a1));***FOCUS***
805         l1.add("hij");
806         System.out.println(l1);Output--->[abc, def, ghi, hij]
807
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:11 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 35 | new 37 | new 38 | SPPrint

```
808
809 Code Example: Converting "generic List" to Array
810         List<String> l=new ArrayList<>();
811         l.add("123");
812         l.add("456");
813         l.add("789");
814         String[] sl=new String[l.size()];
815         l.toArray(sl);
816
817 Code Example: Converting "normal List" to Array
818         List l=new ArrayList<>();
819         l.add("abc");
820         l.add(true);
821         l.add(10);
822         Object[] s1 =l.toArray();
823         System.out.println(s1);
824
825 Collections.swap(list,initial_index,final_index)-->
826     Code Example:
827         List l1=new ArrayList<>();
828         l1.add("element1");
829         l1.add("element2");
830         l1.add("element3");
831         l1.add("element4");
832         System.out.println(l1);//[element1, element2, element3, element4]
833         Collections.swap(l1, 1, 3);
834         System.out.println(l1);//[element1, element4, element3, element2]
835
836 sublist(intial_index,final_index+1)-->
837     Code Example:
838         List l1=new ArrayList<>();
839         l1.add("element1");
```

ormal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:12 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPFRC

```
825 Collections.swap(list,initial_index,final_index)-->
826     Code Example:
827         List l1=new ArrayList<>();
828         l1.add("element1");
829         l1.add("element2");
830         l1.add("element3");
831         l1.add("element4");
832         System.out.println(l1);//[element1, element2, element3, element4]
833         Collections.swap(l1, 1, 3);
834         System.out.println(l1);//[element1, element4, element3, element2]
835
836 sublist(intial_index,final_index+1)-->
837     Code Example:
838         List l1=new ArrayList<>();
839         l1.add("element1");
840         l1.add("element2");
841         l1.add("element3");
842         l1.add("element4");
843         List l2=l1.subList(1, 4);
844         System.out.println(l2); //[element2, element3, element4]
845
846 get(index):
847     List l1=new ArrayList<>();
848         l1.add("element1");
849         l1.add("element2");
850         l1.add("element3");
851         l1.add("element4");
852         System.out.println(l1.get(1));// element2
853
854  Summarize all ArrayList functions:
855                 add(element)
856                 add(position,element)
```

ormal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:12 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
Summarize all ArrayList functions:
                add(element)
                add(position,element)
                addAll(list)
                remove(index)
                remove(element)
                removeAll(list)
                contains(element)
                containsAll(list)
                retainAll(list)
                set(index,element)
                clear()
                size()
                isEmpty()
                converting array to  list
                converting list to array(2)
                Collections.swap(list,initial_index,final_index)
                sublist(intial_index,final_index+1)
                get(index).

Main difference between ArrayList and Vector :Vector is synchronized(one thread at a time)
                                               ArrayList is non synchronized(multiple thread at a time)

                ******Cursors******
we can read collection data using 3 ways
1.forEach loop
2.get()
3.Cursors ---Enumeration Interface
          ---Iterator Interface
          ---ListIterator Interface

ENUMERATIONS:
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:12 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt

```
                ******Cursors******
we can read collection data using 3 ways
1.forEach loop
2.get()
3.Cursors ---Enumeration Interface
          ---Iterator Interface
          ---ListIterator Interface

ENUMERATIONS:
1.introduced version 1.0---->applicable to legacy classes(eg.Vector,stack)
2.It has 2 methods:hasMoreElements(),nextElement()
3.only read operation is applicable
4.forward cursors

Code Example:
        Vector<String> v=new Vector<String>();
        v.add("element1");
        v.add("element2");
        v.add("element3");
        v.add("element4");
        //reading data by using normal version
        Enumeration e1=v.elements();
        while(e1.hasMoreElements()) {
            String s=(String)e1.nextElement();//***type casting is imp***
            System.out.println(s);
        }

        //reading data by using generic version
        Enumeration<String> e2=v.elements();//**focus**
        while(e2.hasMoreElements()) {**************WHILE LOOP USED
            String s=e2.nextElement();
            System.out.println(s);
```

ormal text file length : 53,304 lines : 1,609 Ln : 275 Col : 1 Sel : 2 | 1 Windows (CR LF) UTF-8 INS

Type here to search 14°C Cloudy 3:13 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

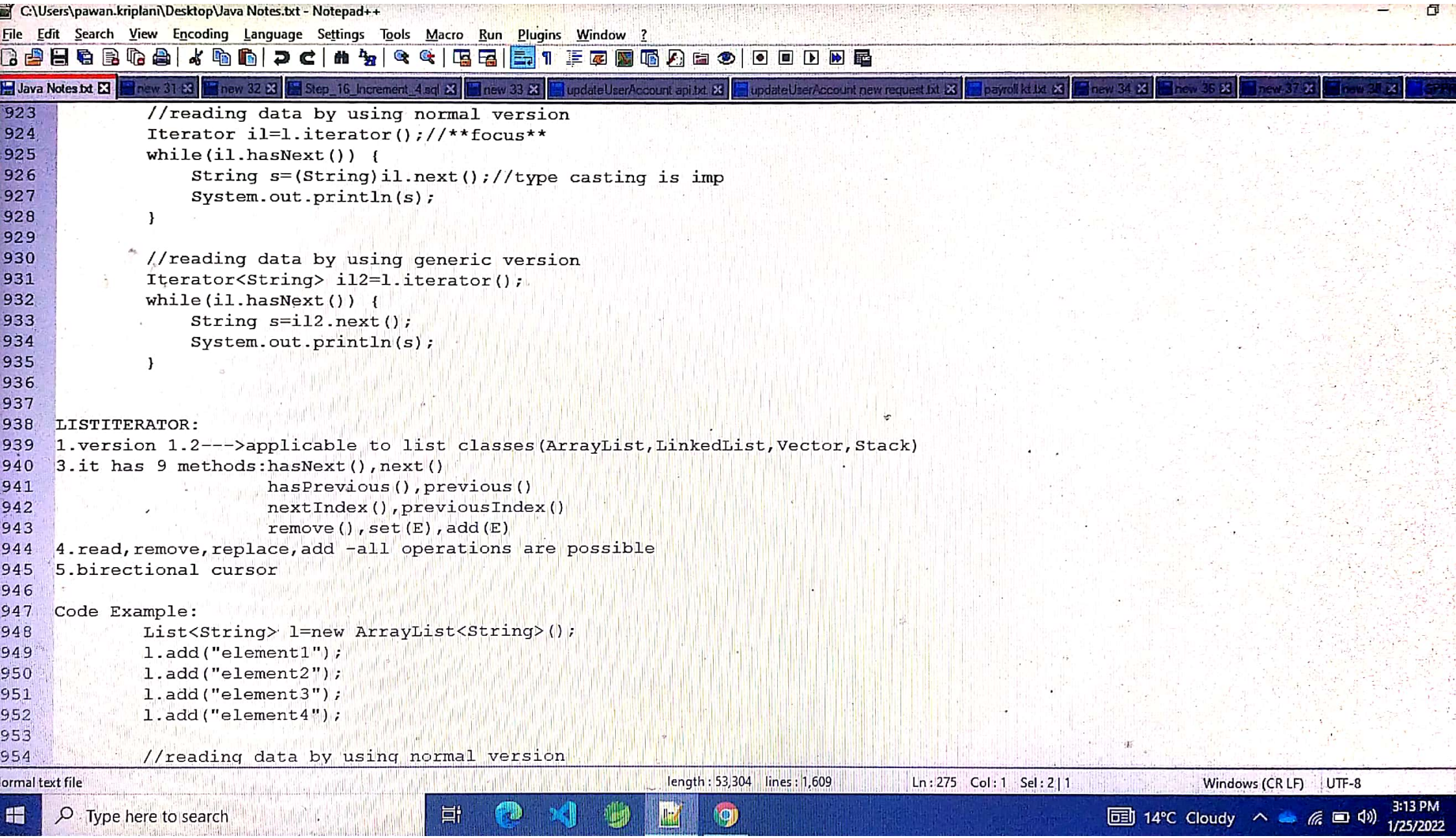
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt

```
892         Vector<String> v=new Vector<String>();
893         v.add("element1");
894         v.add("element2");
895         v.add("element3");
896         v.add("element4");
897         //reading data by using normal version
898         Enumeration e1=v.elements();
899         while(e1.hasMoreElements()) {
900             String s=(String)e1.nextElement();//***type casting is imp***
901             System.out.println(s);
902         }
903
904         //reading data by using generic version
905         Enumeration<String> e2=v.elements();//**focus**
906         while(e2.hasMoreElements()) {*************WHILE LOOP USED
907             String s=e2.nextElement();
908             System.out.println(s);
909         }
910
911 ITERATOR:
912 1.introduced version 1.2---->applicable to ALL classes
913 2.It has 3 methods:hasNext(),next(),remove()
914 3.read and remove operation is allicable
915 4.forward cursors
916
917         List<String> l=new ArrayList<String>();
918         l.add("element1");
919         l.add("element2");
920         l.add("element3");
921         l.add("element4");
922
923         //reading data by using normal version
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8

Type here to search | 14°C Cloudy | 3:13 PM 1/25/20



C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37

```
954         //reading data by using normal version
955     ListIterator i1=l.listIterator();
956     while(i1.hasNext()) {
957         String s=(String)i1.next();
958         System.out.println(s);
959     }
960
961     while(i1.hasPrevious()) {
962         String s=(String)i1.previous();
963         System.out.println(s);
964     }
965
966         //reading data by using generic version
967     ListIterator<String> i2=l.listIterator();
968     while(i2.hasNext()) {
969         String s= i2.next();
970         System.out.println(s);
971     }
972
973     while(i2.hasPrevious()) {
974         String s=i2.previous();
975         System.out.println(s);
976     }
977
978 Note:if we directly use previous without forward ...list will not be itearted
979
980 Adding,Removing and Replacing:****FOCUS ON REMOVING AND REPLACING SYNTAX
981         List<String> l=new ArrayList<String>();
982         l.add("element1");
983         l.add("element2");
984         l.add("element3");
985         l.add("element4");
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 INS

Type here to search | 14°C Cloudy | 3:14 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
    ListIterator<String> i1=l.listIterator();
    i1.add("element0");
    while(i1.hasNext()) {
        String s=i1.next();
        if(s.equals("element2")) {
            i1.remove();
        }
        if(s.equals("element4")) {
            i1.set("elementFOUR");
        }
    }
    System.out.println(l);//[element0, element1, element3, elementFOUR]


Code Example:
    public static void main(String[] args) {
        List<String> list=new ArrayList<>();
        list.add("hello1");
        list.add("hello2");
       ,ListIterator<String> iteartor=list.listIterator();
        iteartor.add("extra");

        while(iteartor.hasNext()) {
            String s=iteartor.next();
            System.out.println(s);
        }
    }
    output-->hello1
             hello2
    Note:is wale example me "extra" print ni hua
                ********************************VECTOR********************
1.introduced in 1.0 v
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8

Type here to search | 14°C Cloudy | 3:15 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount.api.txt | updateUserAccount new request.txt | payroll hr.txt | new 34 | new 36 | new 37 | new 38 | new 39

```
1017                ****************************VECTOR*******************
1018  1.introduced in 1.0 v
1019  2.only one thread can be accessed at one time
1020  3.Constructors:
1021      public vector()
1022          default size :10
1023          new capacity=20(every time it become 2X)
1024      public vector(int user_capacity)
1025          new Vector(7)
1026          new capacity=14
1027      public vector(int user_capacity,int increment_value)
1028          new Vector(4,8)
1029          new capacity=12
1030      public vector(Collections)
1031
1032      eg:
1033          Vector l=new Vector<String>(2,4);
1034          System.out.println(l.capacity());//output-->2
1035          l.add("element1");
1036          l.add("element2");
1037          l.add("element3");
1038          System.out.println(l.capacity());//output-->6
1039
1040      eg:
1041          Vector v=new Vector<String>(2,4);
1042          v.add("element1");
1043          v.add("element2");
1044          v.add("element2");
1045          ArrayList list=new ArrayList<>(v);***focus***
1046          System.out.println(list);
1047
1048
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 INS

Type here to search | 14°C Cloudy | 3:15 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
1049                                    **********STACK***********
1050  1.it follows LIFO(Last in first out)
1051  2.child class of vector
1052  3.methods: E push(E)
1053             E pop()
1054             E peek()
1055         boolean isEmpty()
1056             int search(Object)
1057  4.Null allowed,Dupliactes allowed,Insertion order preserved ,heterogenous data
      allowed,cursors-Enumeration,iterator,ListIterator
1058
1059  Code example:
1060          Stack<String> s=new Stack();
1061          s.push("element1");
1062          s.push("element2");
1063          s.push("element3");
1064          //it follows lifo
1065          //consider adding elements in bucket
1066          //positioning starts from 1
1067          System.out.println(s);//[element1, element2, element3]*****dekho ulte print nii ho rhe h..
1068          System.out.println(s.search("element1"));//3
1069          System.out.println(s.search("element2"));//2
1070          System.out.println(s.search("element3"));//1
1071
1072          s.pop();//removes element present at top of bucket i.e element3
1073          System.out.println(s);//[element1, element2]
1074          System.out.println(s.peek());;//gives element present at tops i.e element2
1075          System.out.println(s);
1076          System.out.println(s.isEmpty());//false
1077          s.clear();
1078          System.out.println(s.isEmpty());//true
1079
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:15 PM 1/25/2022

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37

```
1080                    ******LINKEDLIST******
1081  1.introduce in 1.2 version
1082  2.heterogenous data
1083  3.null values allowed
1084  4.duplicates allowed
1085  5.insertion order preserved
1086  6.non-sync
1087  7.underlined data structure is "Double linked list"
1088  8.iterator and list-iterator  can be used
1089
1090  Constructors:
1091  public LinkedList()
1092  public LinkedList(Collection c)
1093
1094  methods:add(E),add(index,E),addFirst(E),addLast(E),removeFirst(),removeLast()....rest all other lke ArrayList
1095  CodeExample:
1096          LinkedList<String> list=new LinkedList<>();
1097          list.add("element1");
1098          list.add("element2");
1099          list.add("element3");
1100          list.add("element4");
1101          System.out.println(list);//[element1, element2, element3, element4]
1102          list.addFirst("elementFIrst");//"FOCCCCUUUSSSS"***
1103          list.addLast("elementLast");//"FOCCCCUUUSSSS"
1104          System.out.println(list);//[elementFIrst, element1, element2, element3, element4, elementLast]
1105          list.add(2,"extraelement");
1106          System.out.println(list);
1107          list.set(3,"REPLACEDelement");//[elementFIrst, element1, extraelement, REPLACEDelement, element3, element4,
          elementLast]
1108          System.out.println(list);
1109          list.clear();
1110          System.out.println(list.isEmpty());//true
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:16 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
1112 ArrayList vs LinkedList:
1113                 ArrayList                                   LinkedList
1114 1.It uses  array                                           It uses  "doubly linked list"
1115 2.manipulation is slow bcoz adding and removing elements    manipulation is fast
1116     will shift elements
1117 3. act as a list ecause it implements List only.           LinkedList class can act as a list and queue both because it
1118                                                            implements List and Deque interfaces.
1119 4. better for storing and accessing data                     better for manipulating data
1120
1121                 ******************Collection Sorting***************
1122 1.conditon for sorting is -data should be homogenous
1123                           -implements Comparable interface
1124 2.String ,All wrapper class has implemented have implemented comparable interface
1125
1126 Collections.sort(E) is used to sort data....this method internally uses compareTo() method
1127
1128 here we have 3 cases:
1129 case1:proper sorting of String (as it already has implemented comparable interface)
1130 case2:for heterogenous data ClassCastException (CCE) will be thrown
1131 case3:for null values in collection...NullPointerException is thrown
1132
1133 CAse1:CodeExample
1134         ArrayList list=new ArrayList<>();
1135         list.add("Ratan");
1136         list.add("Durga");
1137         list.add("Sravya");
1138         list.add("Anjaan");
1139         System.out.println(list);[Ratan, Durga, Sravya, Anjaan]
1140         Collections.sort(list);
1141         System.out.println(list);[Anjaan, Durga, Ratan, Sravya]
1142
1143 Case2:CodeExample
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:16 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
1143  Case2:CodeExample
1144          ArrayList list=new ArrayList<>();
1145          list.add(1);
1146          list.add(10.5);
1147          list.add("Sravya");
1148          Collections.sort(list);//ClassCastException
1149
1150  case3:CodeExample
1151          ArrayList list=new ArrayList<>();
1152          list.add("Ratan");
1153          list.add("Durga");
1154          list.add(null);
1155          list.add("Anjaan");
1156          Collections.sort(list);
1157
1158  Very Important Question.if we want to Sort Student data on basis of id or name then ?
1159  Answer:CASE1:sorting data using name(String)
1160
1161  public class Student implements Comparable<Student> {
1162      int id;
1163      String name;
1164      public Student(int id,String name) {
1165          this.id=id;
1166          this.name=name;
1167      }
1168
1169      @Override
1170      public int compareTo(Student o) {
1171          return this.name.compareTo(o.name);//String already implements comparable interface
1172      }
1173  }
1174
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8

Type here to search | 14°C Cloudy | 1/25/

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37

```
1175      public static void main(String[] args) {
1176          List<Student> list=new ArrayList<>();
1177          list.add(new Student(222, "pawan"));
1178          list.add(new Student(111, "ayushi"));
1179          list.add(new Student(123, "koshal"));
1180
1181
1182          Collections.sort(list);
1183          for(Student s:list) {
1184              System.out.println(s.id+"  "+s.name);
1185          }
1186      }
1187          output:
1188          111  ayushi
1189          123  koshal
1190          222  pawan
1191
1192  CASE2:sorting data using id
1193
1194  @Override
1195      public int compareTo(Student o) {
1196          if(this.id==o.id)
1197              return 0;
1198          else if(this.id>o.id)
1199              return 1;
1200          else return -1;
1201      }
1202
1203      output :
1204      111  ayushi
1205      123  koshal
1206      222  pawan
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8

Type here to search | 14°C Cloudy

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
1208 Problems occured in comparable sorting:
1209 1.mixing normal class code and sorting logic(compareTo())
1210 2.only sorting can be done using one property at a time either id or name ...
1211
1212 To Overcome these problems we use "Comparator "
1213 1.separate logic
1214 2.multiple property sorting
1215
1216 Code Example of Comparator:
1217     //Student class
1218        public class Student {
1219            int id;
1220            String name;
1221            public Student(int id, String name) {
1222                this.id=id;
1223                this.name=name;
1224            }
1225        }
1226
1227     //separate code if we want to sort by id
1228 public class CompareById implements Comparator<Student>{
1229     @Override
1230     public int compare(Student o1, Student o2) {
1231         if(o1.id==o2.id)
1232             return 0;
1233         else if(o1.id>o2.id)
1234             return 1;
1235         else return -1;
1236     }
1237 }
1238     //separate code if we want to sort by name
1239 public class CompareByName implements Comparator<Student> {
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:17 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16 :Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPRING

```
1238      //separate code if we want to sort by name
1239  public class CompareByName implements Comparator<Student> {
1240      @Override
1241      public int compare(Student o1, Student o2) {
1242          return o1.name.compareTo(o2.name);
1243      }
1244
1245  }
1246
1247  //Main class
1248  public static void main(String[] args) {
1249          List<Student> list=new ArrayList<>();
1250          list.add(new Student(123, "pawan"));
1251          list.add(new Student(222, "ayushi"));
1252          list.add(new Student(111, "koshal"));
1253          Collections.sort(list, new CompareById());*********FOCUS**********
1254          for(Student s:list) {
1255              System.out.println(s.id+"  "+s.name);            output:111  koshal
1256          }                                                              123  pawan
1257                                                                         222  ayushi
1258          Collections.sort(list, new CompareByName());
1259          for(Student s:list) {
1260              System.out.println(s.id+"  "+s.name);            output:222  ayushi
1261          }                                                              111  koshal
1262      }                                                                  123  pawan
1263
1264
1265                                          Comparable VS Comparator
1266  Comparable                                          |                          Comparator
1267  1.java.lang                                         |java.util
1268  2.Can be used with String and all wrapper class     |used mostly for Customization logic eg Employee,Student
1269  3.Sorting logic mix with code                       |separate code
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:17 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

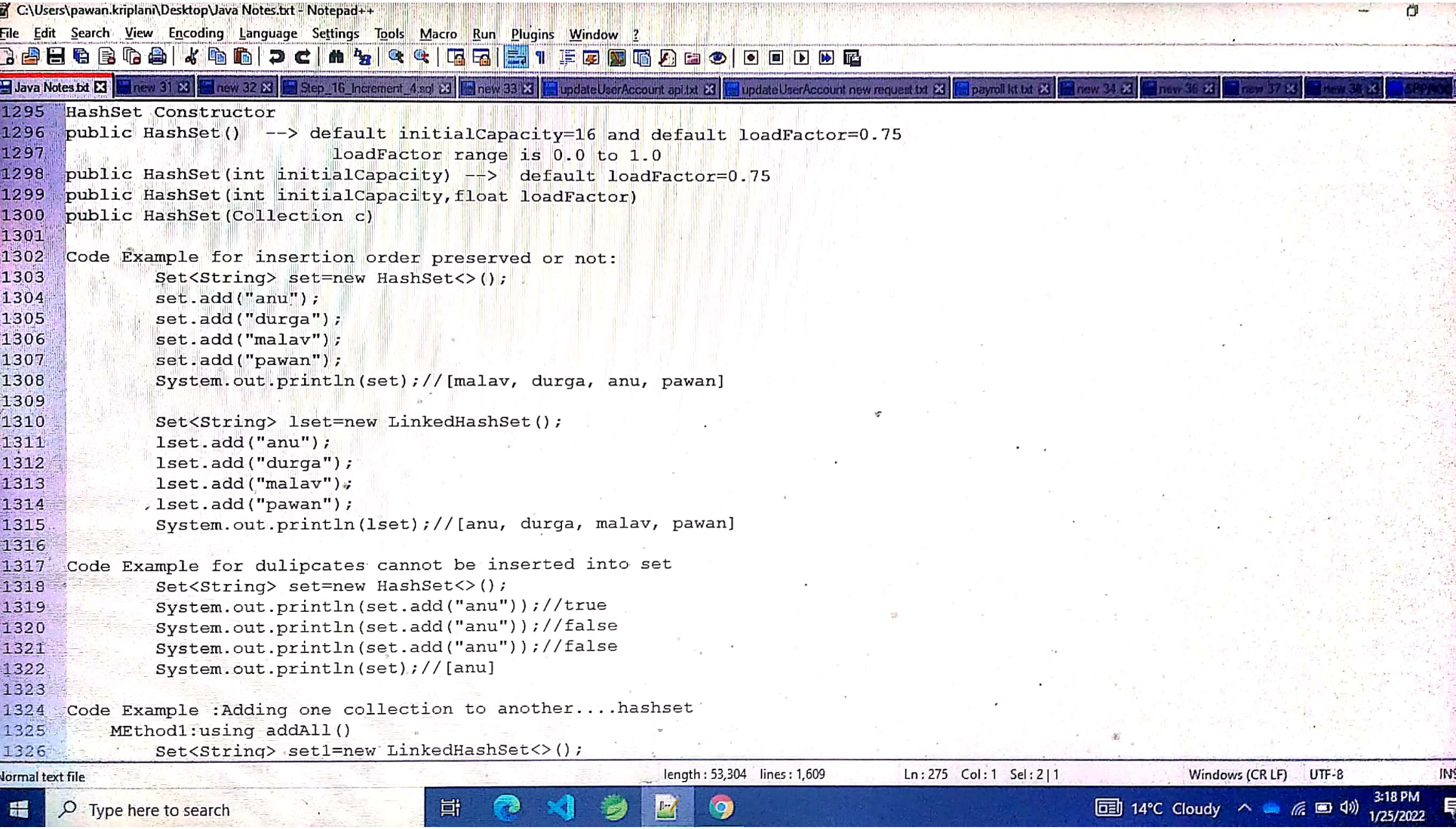
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROC

```
1265                                        Comparable VS Comparator
1266  Comparable                                                |                          Comparator
1267  1.java.lang                                               |java.util
1268  2.Can be used with String and all wrapper class           |used mostly for Customization logic eg Employee,Student
1269  3.Sorting logic mix with code                             |separate code
1270  4. only one property sorting implementation at a time     |multiple property like in above we did with name and id
1271  5.uses CompareTo(Object o) method                         |compare(object o1,object o2)
1272  6.Collections.sort(list)                                  |Collections.sort(list,new Class_name_for_comaprision)
1273
1274                                  *****SET*****
1275
1276                                  Iterable(I) 1.2
1277                                  Collection(I) 1.2
1278                                      Set(I)
1279                                  /           \
1280                      HashSet(c)  1.2              SortedSet(I) 1.2
1281                  LinkedHashSet(C) 1.4             NavigableSet(I) 1.6
1282                                                   TreeSet(C) 1.2
1283
1284  main difference b/w HashSet and LinkedHashSet->HashSet insertion order not preserved but LinkedHashSet insertion order
      preserved
1285  HashSet Characteristics:
1286  1.version 1.2
1287  2.heterogenous
1288  3.null allowed
1289  4.duplicated not allowed
1290  5.insertion order not preserved
1291  6.non-sync
1292  7.underlined data structure is "hashtable"
1293  8.iterator cursor
1294
1295  HashSet Constructor
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:18 PM 1/25/2022



C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPROG

```
1324  Code Example :Adding one collection to another....hashset
1325      MEthod1:using addAll()
1326          Set<String> set1=new LinkedHashSet<>();
1327          set1.add("element1");
1328          set1.add("element2");
1329          Set<String> set2=new LinkedHashSet<>();
1330          set2.addAll(set1);
1331          set2.add("element3");
1332          set2.add("element4");
1333          System.out.println(set2);//[element1, element2, element3, element4]
1334
1335      Method2:using constructor:
1336          Set<String> set1=new LinkedHashSet<>();
1337          set1.add("element1");
1338          set1.add("element2");
1339          Set<String> set2=new LinkedHashSet<>(set1);
1340          set2.add("element3");
1341          set2.add("element4");
1342          System.out.println(set2);//[element1, element2, element3, element4]
1343
1344                              ****TreeSet*****
1345  TreeSet comes from interface SortedSet so for sorting we data should be homogenous.so Treeset can allow only Homogenous data
1346  If heterogenous data is added it will throw ClassCastException
1347  Also null values cannot be added bcoz it will throw NullPointerException
1348  Underlined Data Structure is "Balance Tree"
1349
1350  Constructors:
1351  TreeSet()--->elements are order acc to natural orderng of elements
1352  TreeSet(Collection c)--->elements from collection are order acc to natural orderng of elements
1353  TreeSet(Comparator comparator)--->elements are order acc to comparator orderng of elements
1354  TreeSet(SortedSet s)
1355
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:18 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPP

```
1356 CodeExample:y b acha tarika h ki data ko sort krna hoto TreeSet use krlo
1357
1358         TreeSet<String> tset1=new TreeSet<>();
1359         tset1.add("durga");
1360         tset1.add("sravya");
1361         tset1.add("pawan");
1362         tset1.add("anu");
1363         System.out.println(tset1);//elements sorted in natural order [anu, durga, pawan, sravya]
1364
1365         TreeSet<Integer> tset2=new TreeSet<>();
1366         tset2.add(3);
1367         tset2.add(45);
1368         tset2.add(1);
1369         tset2.add(55);
1370         System.out.println(tset2);//[1, 3, 45, 55]
1371
1372 Sorting in reverse Order:
1373         TreeSet<String> tset1=new TreeSet<>((s1,s2)->s2.compareTo(s1));
1374                                 Or
1375         TreeSet<String> tset1=new TreeSet<>(Collections.reverseOrder());
1376         tset1.add("durga");
1377         tset1.add("sravya");
1378         tset1.add("pawan");
1379         tset1.add("anu");
1380         System.out.println(tset1);//[sravya, pawan, durga, anu]
1381
1382 Sorting of Employee Class objects on basis of name :
1383 Case1:here it will throw error common mistake
1384
1385     //Employee Class
1386     public class Employee {
1387         int id;
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8

Type here to search | 14°C Cloudy | 3:19 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

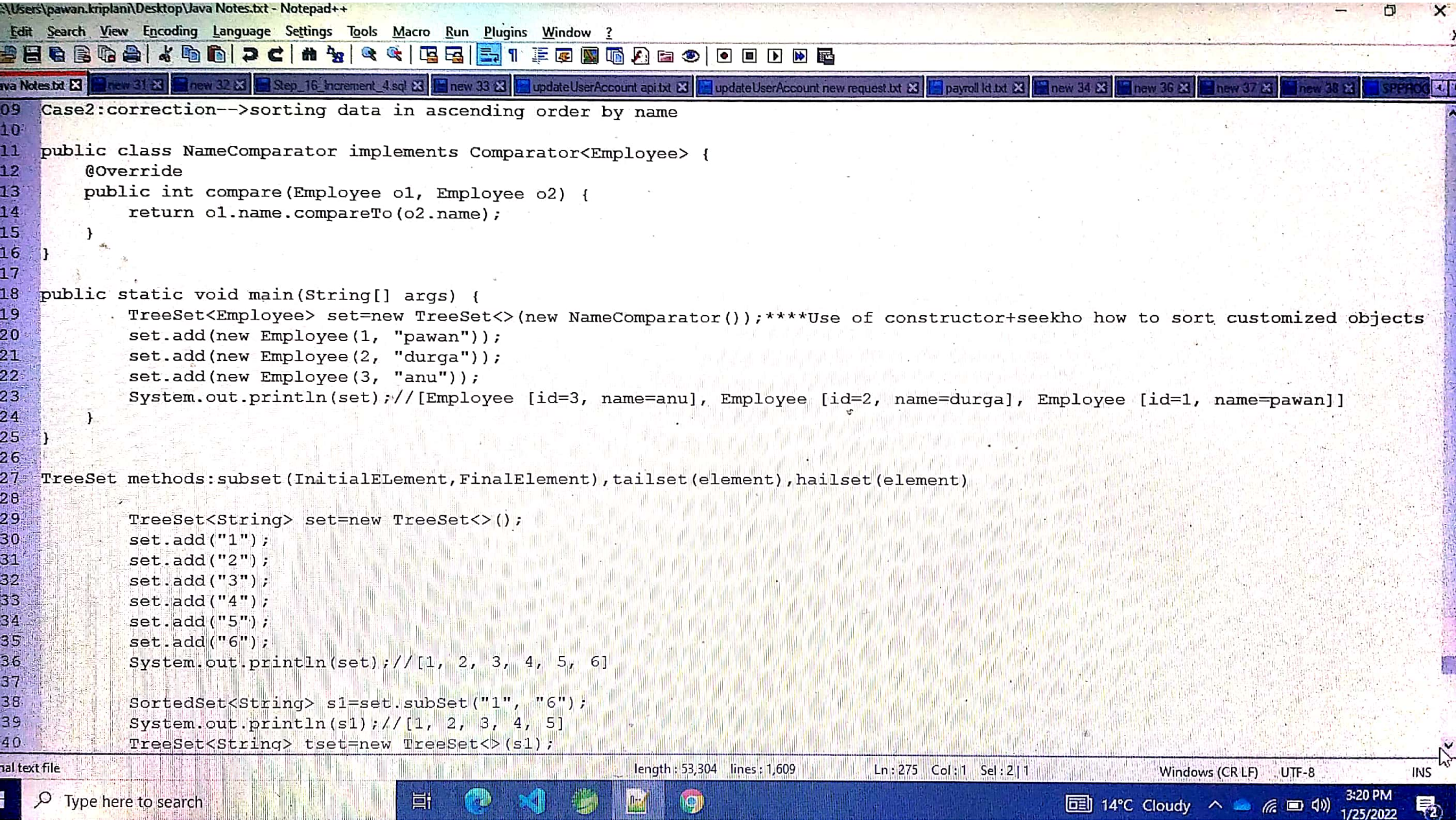
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPRHOC

```
1382 Sorting of Employee Class objects on basis of name :
1383 Case1:here it will throw error common mistake
1384
1385     //Employee Class
1386     public class Employee {
1387         int id;
1388         String name;
1389
1390         public Employee(int id, String name) {
1391             super();
1392             this.id = id;
1393             this.name = name;
1394         }
1395         @Override
1396         public String toString() {
1397             return "Employee [id=" + id + ", name=" + name + "]";
1398         }
1399     }
1400
1401     public static void main(String[] args) {
1402         TreeSet<Employee> set=new TreeSet<>();***ClassCastException bcoz kis basis p sorting kri jae ??
1403         set.add(new Employee(1, "pawan"));
1404         set.add(new Employee(2, "durga"));
1405         set.add(new Employee(3, "anu"));
1406         System.out.println(set);
1407     }
1408
1409 Case2:correction-->sorting data in ascending order by name
1410
1411 public class NameComparator implements Comparator<Employee> {
1412     @Override
1413     public int compare(Employee o1, Employee o2) {
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8 | INS

Type here to search | 14°C Cloudy | 3:19 PM 1/25/2022



C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38

```
1437
1438        SortedSet<String> s1=set.subSet("1", "6");
1439        System.out.println(s1);//[1, 2, 3, 4, 5]
1440        TreeSet<String> tset=new TreeSet<>(s1);
1441        tset.add("asda");
1442        System.out.println(tset);//[1, 2, 3, 4, 5, asda]
1443
1444        SortedSet<String> s2=set.headSet("3");
1445        System.out.println(s2);//[1, 2]-->3 not included
1446        TreeSet<String> tset2=new TreeSet<>(s2);
1447        System.out.println(tset2);//[1, 2]
1448
1449        SortedSet<String> s3=set.tailSet("3");
1450        System.out.println(s3);//[3, 4, 5, 6]-->3 included
1451        TreeSet<String> tset3=new TreeSet<>(s3);
1452        System.out.println(tset3);//[3, 4, 5, 6]
1453
1454
1455                              *******Cloning and Serialization for Collection********
1456  Lecture 37 to 40
1457
1458                              ********Map Interface************
1459  Root Interface of Map is Map
1460  Collection is used to store one object at time
1461  Map is used to store 2 objects at same time in key,value pair
1462
1463                                        Map(I) 1.2
1464                                HashMap(C) 1.2          SortedMap(I) 1.2
1465                                LinkedHashMap(C) 1.4    NavigableMap(I) 1.6
1466                                                        TreeMap(C)1.2
1467
1468  Main difference b/w HashMap and LinkedHashMap? HashMap insertion order is not preserved and LinkedHashMap is  preserved
```

Normal text file | length : 53,304 lines : 1,609 | Ln : 275 Col : 1 Sel : 2 | 1 | Windows (CR LF) UTF-8

Type here to search | 14°C Cloudy | 3:20 PM 1/25/2022

C:\Users\pawan.kriplani\Desktop\Java Notes.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Java Notes.txt | new 31 | new 32 | Step_16_Increment_4.sql | new 33 | updateUserAccount api.txt | updateUserAccount new request.txt | payroll kt.txt | new 34 | new 36 | new 37 | new 38 | SPPBOG

```
1468 Main difference b/w HashMap and LinkedHashMap? HashMap insertion order is not preserved and LinkedHashMap is  preserved
1469 Note:Key must be unique ...Value can be duplicate eg username(key) ,password(value) pair
1470
1471 HASHMAP:
1472 1.Version 1.2
1473 2.Heterogenous data
1474 3.Null allowed
1475 4.Non sync
1476 5.insertion order not preserved
1477 6.HashTable
1478 7.key:unique,Value:duplicate
1479
1480 Constructor:
1481 HashMap()-->initialCapacity=16 and loadFactor=0.75
1482 HashMap(int initialCapacity) loadFactor=0.75
1483 HashMap(int initialCapacity,float loadFactor)
1484 HashMap(Map<k,v>)
1485
1486 Methods: Set keySet()**Focus**
1487        , Collection values()**Focus**
1488          Set<Entry<k,v>> entrySet()**Focus**
1489     CodeExample
1490         HashMap<Integer, String> map1=new HashMap<>();
1491         map1.put(111, "anu");
1492         map1.put(222, "durga");
1493         map1.put(333, "sravya");
1494         map1.put(4, "ratan");
1495         map1.put(6, "durga");
1496         map1.put(5, "pawan");
1497         //Insertion order not preserved
1498         System.out.println(map1);//{4=durga, 5=durga, 6=durga, 333=durga, 222=durga, 111=durga}
1499
```

Normal text file | length : 53,304 | lines : 1,609 | Ln : 275 | Col : 1 | Sel : 2 | 1 | Windows (CR LF) | UTF-8 | INS

Type here to search | 14°C Cloudy | 3:20 PM 1/25/2022