Lab3:

## Remove array element

public void remove(int a[],int index){ //by shifting left  
 if(index>=0 && index<a.length){  
 int i=index;  
 while (i<a.length-1){  
 a[i]=a[i+1];  
 i++;  
 }  
 a[i]=0;  
 }  
}

## Insertion Sort

public void insertionsort(int a[]){  
 for (int i=1;i<a.length;i++){ //تنتقل على جميع العناصر ومن 1 علشان نقارن مع الي قبل  
 int cur=a[i];  
 int j=i;  
 while(j>0&&cur<a[j-1]) //shifting loop  
 {  
 a[j]=a[j-1];  
 j--;  
 }  
 a[j]=cur;  
 }  
}

## Sequential search

public int seqsearch(int a[],int value){  
 int i=0;  
 int index=-1;  
 boolean flag=false;  
 while(i<a.length&&!flag){ //or flag==false  
 if(a[i]==value){  
 flag=true;  
 index=i;  
 }  
 i++;  
 }  
 return index;  
}

## java.util Methods for Arrays

* Arrays.equals(A,B)
* Arrays.fill(A,x)
* Arrays.copyOf(A,n)
* Arrays.coptOfRange(A,s,t)
* Arrays.sort(A)
* Arrays.binarySearch(A,x)
* import java.util.Arrays;
* /\*System.out.println(Arrays.equals(o.q,o.f));  
    
   o.f=Arrays.copyOf(o.q,o.q.length);  
    
   Arrays.fill(o.f,1);  
   System.out.println(Arrays.toString(o.f));  
    
   System.out.println(Arrays.toString(o.q));  
   Arrays.sort(o.q);  
   System.out.println(Arrays.binarySearch(o.q,114));\*/

## Fill array with Random Number

import java.util.Random;

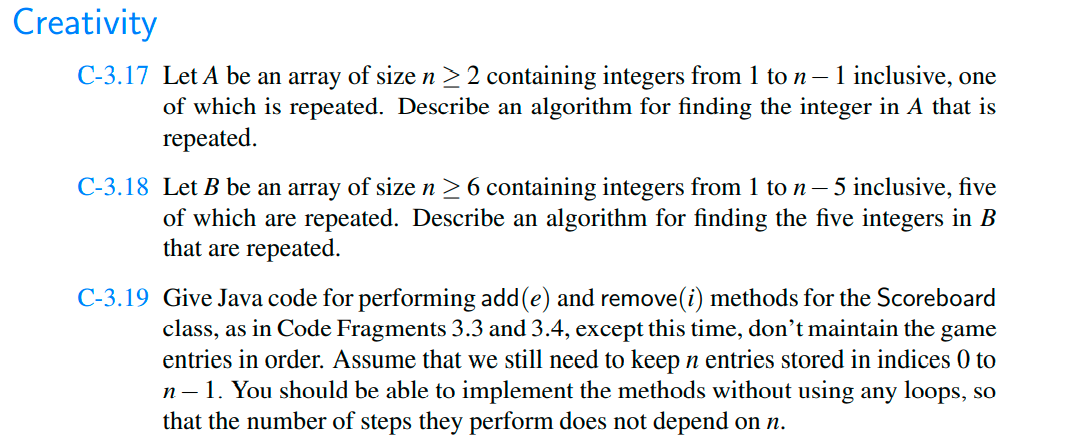
public void fillrandom(){  
 Random r=new Random();  
 r.setSeed(System.*currentTimeMillis*());  
 for(int i=0;i<q.length;i++){  
 q[i]=r.nextInt(10);  
 }  
}

## Convert String to char array

toCharArray

new String(char\_array)

Lab3 Tasks:



We know [sum of first n-1 natural numbers](https://www.geeksforgeeks.org/program-find-sum-first-n-natural-numbers/)is (n – 1)\*n/2. We compute sum of array elements and subtract natural number sum from it to find the only missing element.

3.17:

public class Tasks3 {  
  
  
 int num = 1;  
  
 public int repeate(int []N) {  
  
 int index=0;  
 boolean flag=false;  
  
  
 for(int j=0;j<N.length;j++) {  
 int i=j+1;  
 while (i < N.length && !flag) { //or flag==false  
 if (N[i] == N[j]) {  
 flag = true;  
 num++;  
 }  
 i++;  
 }  
  
 if (flag = true){index++;}  
 else {num=0;}  
 }  
 return index;  
  
 }  
  
 static int findRepeating(int[] arr, int n)  
 {  
 // Find array sum and subtract sum  
 // first n-1 natural numbers from it  
 // to find the result.  
  
 int sum = 0;  
 for (int i = 0; i < n; i++)  
 sum += arr[i];  
 return sum - (((n - 1) \* n)/2 );  
 }  
  
  
 public static void main(String[] args) {  
 Tasks3 o=new Tasks3();  
 int[] A = {1, 1};  
 int[] B = {5,5,5,5,5};  
  
  
 System.*out*.println("there are "+o.num+  
 " of integers are repeated "+o.repeate(A)+" times in the array:");  
  
 System.*out*.println(o.*findRepeating*(A,A.length));  
  
  
 /\* System.out.println("there are "+o.num+  
 " of integers are repeated "+o.repeate(B)+" times in the array:");  
  
 System.out.println(o.findRepeating(B,B.length));\*/  
  
 }  
}

Output:

there are 1 of integers are repeated 2 times in the array:

1

3.18: Only this method has changed:

static int findRepeating(int[] arr, int n)  
{  
 // Find array sum and subtract sum  
 // first n-1 natural numbers from it  
 // to find the result.  
  
 int sum = 0;  
 for (int i = 0; i < n; i++)  
 sum += arr[i];  
 return sum - (((n - 1) \* n) );  
}

System.*out*.println("there are "+o.num+  
 " of integers are repeated "+o.repeate(B)+" times in the array:");  
  
 System.*out*.println(o.*findRepeating*(B,B.length));  
  
}

Output:

there are 1 of integers are repeated 5 times in the array:

5

The Game : (NOT COMPLETED)

public class Game {  
   
 public class GameEntry{  
 private String name ;  
 private int score;  
  
 public GameEntry(String name, int score) {  
 this.name = name;  
 this.score = score;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getScore() {  
 return score;  
 }  
  
 @Override  
 public String toString() {  
 return "GameEntry{" +  
 "name='" + name + '\'' +  
 ", score=" + score +  
 '}';  
 }  
 }  
   
 public class Scoreboard{  
   
 private int numEntries =0;  
 private GameEntry[] board;  
   
 public Scoreboard(int capacity){  
 board=new GameEntry[capacity];  
 }  
   
 public void add(GameEntry e){  
 int newScore =e.getScore();  
   
 if (numEntries<board.length||newScore>board[numEntries-1].getScore()){  
 if(numEntries<board.length)  
 numEntries++;  
   
 int j=numEntries-1;  
 while(j>0&&board[j-1].getScore()<newScore){  
 board[j]=board[j-1];  
 j--;  
 }  
 board[j]=e;  
 }  
   
 }  
   
   
 public GameEntry remove(int i)throws IndexOutOfBoundsException{  
 if(i<0||i>numEntries)  
 throw new IndexOutOfBoundsException("invaled:"+i);  
   
 GameEntry temp =board[i];  
   
 for(int j=i;j<numEntries-1;j++){  
 board[j]=board[j+1];  
 }  
   
 board[numEntries-1]=null;  
 numEntries--;  
 return temp;  
 }  
   
 }  
   
   
}

not completed ..

Lab4:

## Casting

### Widening

### A widening conversion occurs when a type T is converted into a “wider” type U.

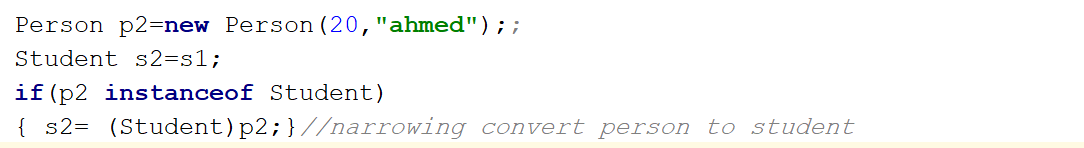
Person p1;  
Student s1 = new Student("Fatima", 21, 555);  
  
p1 = s1; //widining cast  
System.*out*.println(p1);

### Narrowing

* Person p2 = new Person("Suho", 23);  
  if (p2 instanceof Student) {//handling cast exception  
   Student s2 = (Student) p2;//narrowing cast  
   System.*out*.println(s2);  
  }
* //succeed casting
* Person p2 = new Student("Suho", 23, 454);

if (p2 instanceof Student) {//handling cast exception  
 Student s2 = (Student) p2;//narrowing cast  
 System.*out*.println(s2);  
}

### Casting exception



## Generics

Pair objects (Book,Author)(Book,quantity)(label,price)(name,id)

import java.util.Objects;  
public class OldPair {  
  
 Object frist;  
 Object second;  
  
 public OldPair(Object frist, Object second) {  
 this.frist = frist;  
 this.second = second;  
 }  
  
 public Object getFrist() {  
 return frist;  
 }  
  
 public void setFrist(Object frist) {  
 this.frist = frist;  
 }  
  
 public Object getSecond() {  
 return second;  
 }  
  
 public void setSecond(Object second) {  
 this.second = second;  
 }  
  
 @Override  
 public String toString() {  
 return "OldPair{" +  
 "frist=" + frist +  
 ", second=" + second +  
 '}';  
 }  
  
 public static void main(String[] args) {  
 OldPair bookPair=new OldPair("book",11);  
 String x=(String) bookPair.getFrist();  
 int y=(int) bookPair.getSecond();  
 System.*out*.println(bookPair);  
  
 //Student s1=new Student("f",33,122);  
 //Student s2=new Student("s",22,922);  
 //OldPair comp=new OldPair(s1,s2);  
  
 }  
}

* **Generics and arrays**
* **Generic Methods**
* public class Pair<A,B> {  
   A first;  
   B second;  
    
   public Pair(A first, B second) {  
   this.first = first;  
   this.second = second;  
   }  
    
   public A getFirst() {  
   return first;  
   }  
    
   public void setFirst(A first) {  
   this.first = first;  
   }  
    
   public B getSecond() {  
   return second;  
   }  
    
   public void setSecond(B second) {  
   this.second = second;  
   }  
    
   @Override  
   public String toString() {  
   return "Pair{" +  
   "first=" + first +  
   ", second=" + second +  
   '}';  
   }  
    
   public static void main(String[] args) {  
   Pair <String ,Integer>bookpair =new Pair<>("book",11);  
   String x=bookpair.getFirst();  
   System.*out*.println(bookpair);  
   }  
  }

public class GenerkArray<I> {  
  
 I arr[];  
 int num=0;  
  
 public GenerkArray() {  
 arr=(I[])new Object[2];  
 }  
 public void add(I element){  
 if(num<=arr.length-1){  
 arr[num++]=element;  
 }  
 }  
 public void print(){  
 for (int i=0;i<arr.length;i++){  
 System.*out*.println(arr[i]);  
 }  
 }  
  
 public static void main(String[] args) {  
 GenerkArray <Integer> a1=new GenerkArray<>();  
 a1.add(11);  
 a1.add(12);  
 a1.print();  
 GenerkArray <String> a2=new GenerkArray<>();  
 a2.add("aaaaa");  
 a2.add("fffff");  
 a2.print();  
 }  
}

public class GenerkMethod {  
 public static <E> void print(E arr[]){  
 for (int i = 0; i <arr.length ; i++) {  
 System.*out*.println(arr[i]);  
 }  
 }  
  
 public static void main(String[] args) {  
 Integer a[] = {1, 2, 3, 4, 5};  
 String b[]={"hi","bye"};  
  
 *print*(a);  
 *print*(b);  
 }  
}

## Nested Classes

Lab4 Tasks:

1. Write generic method to reverse arrays of any type.
2. Write generic method to merge tow arrays of any type .

public class Task4 {  
  
  
 public static <E>void reverse(E arr[])  
 {  
  
 int n= arr.length-1;  
 for(int i=0;i<arr.length/2;i++){  
 E t= arr[i];  
 arr[i]=arr[n];  
 arr[n]=t;  
 n--;  
 }  
  
 for (int i = 0; i <arr.length ; i++) {  
 System.*out*.println(arr[i]);  
 }  
  
 }  
 public static <E>void merge(E a[],E b[]){  
 E c[]= (E[]) new Object[a.length+b.length];  
  
 int i ;  
  
 for(i=0;i<a.length;i++){  
 c[i]=a[i];  
 }  
  
 for(int j =i, l=0;i<c.length&&l<b.length;j++){  
 c[j]=b[l++];  
 }  
  
 for (int f = 0; f <c.length ; f++) {  
 System.*out*.print(c[f]);  
 }  
  
 System.*out*.println("merging is done");  
 }  
  
  
 public static void main(String[] args) {  
 Integer numbers[]={1,2,3,4,5};  
 String names[]={"a","b","c"};  
 String ppl[]={"D","E","F"};  
  
 *reverse*(numbers);  
 *merge*(names,ppl);  
 }  
}

Output:

5

4

3

2

1

abcDEF merging is done.