**DATA STRUCTURE :**way of organizing , storing , manipulating , retrieving the data .

ADT : Abstract Data Type is conceptual module that define the set of operation and behavior for a data type without specifying the how the operation are implemented or how the data is organize in the memory .

Ex :  
stack , array , linked list ,map .

package day13;

import java.util.\*;

class StackArray {

int arr[], top, MAX;

public StackArray(int MAX) {

this.top = -1;

this.MAX = MAX;

this.arr = new int[MAX];

}

void push(int val) {

if (overflow()) {

System.***out***.println("Stack Overflow");

return;

}

arr[++top] = val;

System.***out***.println("Pushed: " + val);

}

int pop() {

if (isEmpty()) {

System.***out***.println("Stack Underflow");

return -1;

}

return arr[top--];

}

void peek()

{

if(isEmpty())

{

System.***out***.println("stack Empty");

}

else {

System.***out***.println("peek element :"+ arr[top]);

}

}

boolean overflow() {

return top >= MAX - 1;

}

boolean isEmpty() {

return top == -1;

}

}

public class task2 {

public static void main(String args[]) {

Scanner s = new Scanner(System.***in***);

int i=1, MAX = 5;

StackArray stack = new StackArray(MAX);

while(i<=MAX)

{

stack.push(s.nextInt());

i++;

}

stack.push(60); // over flow

System.***out***.println("Popped: " + stack.pop());

System.***out***.println("Popped: " + stack.pop());

stack.peek(); // top element

System.***out***.println("stack empty :"+stack.isEmpty());

}

}

String :

Creation :

package day13;

public class task5 {

public static void main(String args[])

{

String s2;

s2 = "hari";

String s3 = new String("hari");

char[] ch = {'a','a','b','b'};

String s4 = new String(ch);

//String s5 = new String(ch,1,4);

System.***out***.println(s2);

System.***out***.println(s3);

System.***out***.println(s4);

System.***out***.println();

System.***out***.println(s2);

}

}

package day13;

public class task5 {

public static void main(String args[])

{

String s1;

s1 = "hari";

String s2 = "hari";

String s3 = new String("hari");

System.***out***.println(s1==s2);

System.***out***.println(s1==s3);

}

}

true

false

string always check for the existing of data that is assign to it in the string constant .

if the data if available in string constant 2 then string does not create a new space for storing the data instant it just refer to the existing data .

hence == operator check for object reference and not the content . string s1 and s2 evaluate to true

since string s3 create the obj using new key word it forces to create a new space rather then pointing to the existing constant 2 .