Object Diagrams

Q56 Step I

==========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){

}

}

class Demo{

public static void main(String args[]){

Customer c1;

c1=new Customer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q57 Step II

===========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){

}

}

class Demo{

public static void main(String args[]){

Customer c1;

new Customer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q58 Step III

============

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

}

class Demo{

public static void main(String args[]){

Customer c1;

c1=new Customer();

c1.code=1001;

c1.name="Danapala";

}

}

//////////////////////////////////////////////////////////////////////////////

Q59 Step IV

===========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

Customer c2=new Customer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q60 Step VI

===========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

c1.printCustomer(); //1001-Danapala

Customer c2=new Customer();

c2.printCustomer(); //0-null

}

}

//////////////////////////////////////////////////////////////////////////////

Q61 Step V

==========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

c1.printCustomer();

Customer c2=new Customer();

c2.printCustomer();

c1=c2;

c1.printCustomer();

c2.printCustomer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q62 Step VI

===========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

public void setCustomer(int code, String name){

this.code=code;

this.name=name;

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

Customer c2=new Customer();

c2.setCustomer(1002, "Gunapala");

c1.printCustomer();

c2.printCustomer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q63 Step VII

============

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

public void setCustomer(int code, String name){

this.code=code;

this.name=name;

}

public void setCustomer(Customer c1){

this.code=c1.code;

this.name=c1.name;

//setCustomer(c1.code,c1.name); //- OK

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

Customer c2=new Customer();

c1.printCustomer();

c2.printCustomer();

c2.setCustomer(c1);

c1.printCustomer();

c2.printCustomer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q64 Step VIII

=============

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

public void setCustomer(int code, String name){

this.code=code;

this.name=name;

}

public void setCustomer(Customer c1){

this.code=c1.code;

this.name=c1.name;

}

public Customer getCopy(){

return this;

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

Customer c2;

c2=c1.getCopy();

c1.printCustomer();

c2.printCustomer();

System.out.println(c1==c2);

}

}

//////////////////////////////////////////////////////////////////////////////

Q65 Step IX

===========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

public void setCustomer(int code, String name){

this.code=code;

this.name=name;

}

public void setCustomer(Customer c1){

this.code=c1.code;

this.name=c1.name;

}

public Customer getCopy(){

return new Customer(code,name);

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

Customer c2;

c2=c1.getCopy();

c1.printCustomer();

c2.printCustomer();

System.out.println(c1==c2);

}

}

//////////////////////////////////////////////////////////////////////////////

Q66 Step X

==========

class Customer{

int code;

String name;

Customer(int code, String name){

this.code=code;

this.name=name;

}

Customer(){ }

public void printCustomer(){

System.out.println(code+" - "+name);

}

public void setCustomer(int code, String name){

this.code=code;

this.name=name;

}

public void setCustomer(Customer c1){

this.code=c1.code;

this.name=c1.name;

}

public Customer getCopy(){

return new Customer(code,name);

}

public boolean equals(Customer cust){

return this.code==cust.code;

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001,"Danapala");

Customer c2=new Customer(1001,"Danapala");

Customer c3=new Customer(1002,"Gunapala");

Customer c4=c1;

System.out.println(c1==c2);

System.out.println(c1==c3);

System.out.println(c1==c4);

System.out.println(c1.equals(c2));

System.out.println(c1.equals(c3));

System.out.println(c1.equals(c4));

}

}

//////////////////////////////////////////////////////////////////////////////

Q67 Step XI

===========

class Order{

String orderId;

}

class Customer{

int code;

String name;

Order order;

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q68 Step XII

============

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Customer{

int code;

String name;

Order order;

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer();

c1.code=1001;

c1.name="Danapala";

c1.order=new Order("D001");

}

}

//////////////////////////////////////////////////////////////////////////////

Q69 Step XIII

=============

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Customer{

int code;

String name;

Order order;

Customer(int code, String name){

this.code=code;

this.name=name;

}

public void setOrder(Order order){

this.order=order;

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001, "Danapala");

c1.setOrder(new Order("D001");

}

}

//////////////////////////////////////////////////////////////////////////////

Q70 Step XIV

============

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Customer{

int code;

String name;

Order order;

Customer(int code, String name, Order order){

this.code=code;

this.name=name;

this.order=order;

}

public void printCustomer(){

System.out.println(code+" - "+name+" - "+order.orderId);

}

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer(1001, "Danapala", new Order("D001"));

c1.printCustomer();

}

}

//////////////////////////////////////////////////////////////////////////////

Q71 Step XV

===========

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Demo{

public static void main(String args[]){

int[] xr;

xr=new int[5];

Order[] orders;

orders=new Order[5];

}

}

//////////////////////////////////////////////////////////////////////////////

Q72 Step XVI

============

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Demo{

public static void main(String args[]){

int[] xr=new int[5];

Order[] orders=new Order[5];

xr[0]=100;

orders[0]=new Order("D001");

}

}

//////////////////////////////////////////////////////////////////////////////

Q73 Step XVII

=============

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Demo{

public static void main(String args[]){

int[] xr={10,20,30,40,50};

Order[] orders={

new Order("D001"),

new Order("D002"),

new Order("D003"),

new Order("D004"),

new Order("D005"),

};

}

}

//////////////////////////////////////////////////////////////////////////////

Q74 Step XVIII

==============

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Demo{

public static void main(String args[]){

int[] xr=new int[5];

for (int i = 0; i < 5; i++){

xr[i]=i;

}

Order[] orders=new Order[5];

for (int i = 0; i < 5; i++){

orders[i]=new Order("D00"+i);

}

}

}

//////////////////////////////////////////////////////////////////////////////

Q75 Step XIX

============

class Employee{

String name;

double salary;

Employee(String name, double salary){

this.name=name;

this.salary=salary;

}

public void printEmployee(){

System.out.println(name+" - "+salary);

}

public void incrementSalary(double rate){

salary\*=(rate+100)/100;

}

}

class Demo{

public static void main(String args[]){

Employee[] employees={

new Employee("Danapala",10000.0),

new Employee("Gunapala",20000.0),

new Employee("Somapala",30000.0),

new Employee("Siripala",40000.0),

new Employee("Amarapala",50000.0),

};

for(Employee employee : employees){

employee.printEmployee();

}

//Increment salary

for(Employee employee : employees){

employee.incrementSalary(10); //10%

}

System.out.println("------------------");

for(Employee employee : employees){

employee.printEmployee();

}

}

}

//////////////////////////////////////////////////////////////////////////////

Q76 Step XX

===========

class Order{

String orderId;

Order(String orderId){this.orderId=orderId;}

}

class Customer{

int code;

String name;

Order [] orders;

int[] xr;

}

class Demo{

public static void main(String args[]){

Customer c1=new Customer();

c1.code=1001;

c1.name="Danapala";

c1.orders=new Order[]{

new Order("D001"),

new Order("D002"),

new Order("D003"),

new Order("D004"),

new Order("D005"),

};

c1.xr=new int[]{10,20,30,40,50};

}

}

//////////////////////////////////////////////////////////////////////////////

class "Stack"

//////////////////////////////////////////////////////////////////////////////

Q77 Step I

==========

class Stack{

private int[] dataArray;

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

}

}

//////////////////////////////////////////////////////////////////////////////

Q78 Step II

===========

class Stack{

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

}

}

//////////////////////////////////////////////////////////////////////////////

Q79 Step III

============

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

}

}

//////////////////////////////////////////////////////////////////////////////

Q80 Step IV

===========

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

dataArray[nextIndex++]=data;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q81 From Q80

============

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

dataArray[nextIndex++]=data;

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println("\b\b]");

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q82 Step V

==========

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

dataArray[nextIndex++]=data;

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println("\b\b]");

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

s1.pop();

s1.printStack(); //[40, 30, 20, 10]

s1.pop();

s1.printStack(); //[30, 20, 10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q83 From Q82

============

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

dataArray[nextIndex++]=data;

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println("\b\b]");

}

public void pop(){

nextIndex--;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

s1.pop();

s1.printStack(); //[40, 30, 20, 10]

s1.pop();

s1.printStack(); //[30, 20, 10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q84 Step VI

===========

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

dataArray[nextIndex++]=data;

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println("\b\b]");

}

public void pop(){

nextIndex--;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

System.out.println("Size of the stakc : "+s1.isze()); //0

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack();

System.out.println("Size of the stakc : "+s1.isze()); //5

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.clear();

s1.printStack();//[empty]

System.out.println("Size of the stakc : "+s1.isze()); //0

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.pop(); //Prints "Stack is empty.."

for (int i = 0; i < 10; i++){

s1.push(i);

}

s1.printStack(); //[9,8,7,6,5,4,3,2,1,0]

System.out.println("Size of the stakc : "+s1.isze()); //10

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.push(10); //Prints "Stack is full...."

}

}

//////////////////////////////////////////////////////////////////////////////

Q85 From Q84

============

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

if(nextIndex>=dataArray.length){

System.out.println("Stack is full...");

}else{

dataArray[nextIndex++]=data;

}

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println(nextIndex<=0 ? "empty]":"\b\b]");

}

public void pop(){

if(nextIndex<=0){

System.out.println("Stack is empty...");

}else{

nextIndex--;

}

}

public int size(){

return nextIndex;

}

public int capacity(){

return dataArray.length;

}

public void clear(){

nextIndex=0;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10);

System.out.println("Size of the stakc : "+s1.size()); //0

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack();

System.out.println("Size of the stakc : "+s1.size()); //5

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.clear();

s1.printStack();//[empty]

System.out.println("Size of the stakc : "+s1.size()); //0

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.pop(); //Prints "Stack is empty.."

for (int i = 0; i < 10; i++){

s1.push(i);

}

s1.printStack(); //[9,8,7,6,5,4,3,2,1,0]

System.out.println("Size of the stakc : "+s1.size()); //10

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.push(10); //Prints "Stack is full...."

}

}

//////////////////////////////////////////////////////////////////////////////

Q86 Step VII

============

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

public void push(int data){

if(nextIndex>=dataArray.length){

System.out.println("Stack is full...");

}else{

dataArray[nextIndex++]=data;

}

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println(nextIndex<=0 ? "empty]":"\b\b]");

}

public void pop(){

if(nextIndex<=0){

System.out.println("Stack is empty...");

}else{

nextIndex--;

}

}

public int size(){

return nextIndex;

}

public int capacity(){

return dataArray.length;

}

public void clear(){

nextIndex=0;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(5);

System.out.println("Size of the stakc : "+s1.size()); //0

System.out.println("Capacity of the stakc : "+s1.capacity()); //5

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

System.out.println("Size of the stakc : "+s1.size()); //5

System.out.println("Capacity of the stakc : "+s1.capacity()); //5

s1.push(60);

System.out.println("Size of the stakc : "+s1.size()); //6

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.printStack(); //[60, 50, 40, 30, 20, 10]

s1.push(70);

s1.push(80);

s1.push(90);

s1.push(100);

s1.printStack(); //[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

System.out.println("Size of the stakc : "+s1.size()); //10

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.push(200);

s1.printStack(); //[200, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

System.out.println("Size of the stakc : "+s1.size()); //11

System.out.println("Capacity of the stakc : "+s1.capacity()); //20

}

}

//////////////////////////////////////////////////////////////////////////////

Q87 From Q86

============

class Stack{

private int nextIndex;

private int[] dataArray;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

}

private boolean isFull(){

return nextIndex>=dataArray.length;

}

private void createNewArray(){

int[] newArray=new int[2\*dataArray.length];

for(int i=0; i<nextIndex; i++){

newArray[i]=dataArray[i];

}

dataArray=newArray;

}

public void push(int data){

if(isFull()){

createNewArray();

}

dataArray[nextIndex++]=data;

}

public void printStack(){

System.out.print("[");

for (int i = nextIndex-1; i >=0; i--){

System.out.print(dataArray[i]+", ");

}

System.out.println(nextIndex<=0 ? "empty]":"\b\b]");

}

public void pop(){

if(nextIndex<=0){

System.out.println("Stack is empty...");

}else{

nextIndex--;

}

}

public int size(){

return nextIndex;

}

public int capacity(){

return dataArray.length;

}

public void clear(){

nextIndex=0;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(5);

System.out.println("Size of the stakc : "+s1.size()); //0

System.out.println("Capacity of the stakc : "+s1.capacity()); //5

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

System.out.println("Size of the stakc : "+s1.size()); //5

System.out.println("Capacity of the stakc : "+s1.capacity()); //5

s1.push(60);

System.out.println("Size of the stakc : "+s1.size()); //6

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.printStack(); //[60, 50, 40, 30, 20, 10]

s1.push(70);

s1.push(80);

s1.push(90);

s1.push(100);

s1.printStack(); //[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

System.out.println("Size of the stakc : "+s1.size()); //10

System.out.println("Capacity of the stakc : "+s1.capacity()); //10

s1.push(200);

s1.printStack(); //[200, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

System.out.println("Size of the stakc : "+s1.size()); //11

System.out.println("Capacity of the stakc : "+s1.capacity()); //20

}

}

//////////////////////////////////////////////////////////////////////////////

Class Queue

//////////////////////////////////////////////////////////////////////////////

Q88 Step I

==========

class Queue{

private int nextIndex;

private int[] dataArray;

Queue(int size){

dataArray=new int[size];

nextIndex=0;

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue(10);

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

}

}

//////////////////////////////////////////////////////////////////////////////

Q89 From Step I

===============

class Queue{

private int nextIndex;

private int[] dataArray;

Queue(int size){

dataArray=new int[size];

nextIndex=0;

}

public void enQueue(int data){

if(nextIndex>=dataArray.length){

//full

}else{

dataArray[nextIndex++]=data;

}

}

private boolean isEmpty(){

return nextIndex<=0;

}

public void printQueue(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue(10);

q1.printQueue(); //[empty]

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

}

}

//////////////////////////////////////////////////////////////////////////////

Q90 Step II

===========

class Queue{

private int nextIndex;

private int[] dataArray;

Queue(int size){

dataArray=new int[size];

nextIndex=0;

}

private boolean isFull(){

return nextIndex>=dataArray.length;

}

private void createNewArray(){

int[] newArray=new int[2\*dataArray.length];

for(int i=0; i<nextIndex; i++){

newArray[i]=dataArray[i];

}

dataArray=newArray;

}

public void enQueue(int data){

if(isFull()){

createNewArray();

}else{

dataArray[nextIndex++]=data;

}

}

private boolean isEmpty(){

return nextIndex<=0;

}

public void printQueue(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue(10);

q1.printQueue(); //[empty]

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

q1.deQueue();

q1.printQueue(); //[20, 30, 40, 50]

q1.deQueue();

q1.printQueue(); //[30, 40, 50]

}

}

//////////////////////////////////////////////////////////////////////////////

Q91 From Step II

================

class Queue{

private int nextIndex;

private int[] dataArray;

Queue(int size){

dataArray=new int[size];

nextIndex=0;

}

public void deQueue(){

if(isEmpty()){

System.out.println("Queue is empty..");

}else{

for(int i=0; i<nextIndex-1; i++){

dataArray[i]=dataArray[i+1];

}

nextIndex--;

}

}

private boolean isFull(){

return nextIndex>=dataArray.length;

}

private void createNewArray(){

int[] newArray=new int[2\*dataArray.length];

for(int i=0; i<nextIndex; i++){

newArray[i]=dataArray[i];

}

dataArray=newArray;

}

public void enQueue(int data){

if(isFull()){

createNewArray();

}else{

dataArray[nextIndex++]=data;

}

}

private boolean isEmpty(){

return nextIndex<=0;

}

public void printQueue(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue(10);

q1.printQueue(); //[empty]

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

q1.deQueue();

q1.printQueue(); //[20, 30, 40, 50]

q1.deQueue();

q1.printQueue(); //[30, 40, 50]

}

}

//////////////////////////////////////////////////////////////////////////////

Q92 Step III

============

class Queue{

private int nextIndex;

private int[] dataArray;

Queue(int size){

dataArray=new int[size];

nextIndex=0;

}

public void deQueue(){

if(isEmpty()){

System.out.println("Queue is empty..");

}else{

for(int i=0; i<nextIndex-1; i++){

dataArray[i]=dataArray[i+1];

}

nextIndex--;

}

}

private boolean isFull(){

return nextIndex>=dataArray.length;

}

private void createNewArray(){

int[] newArray=new int[2\*dataArray.length];

for(int i=0; i<nextIndex; i++){

newArray[i]=dataArray[i];

}

dataArray=newArray;

}

public void enQueue(int data){

if(isFull()){

createNewArray();

}else{

dataArray[nextIndex++]=data;

}

}

private boolean isEmpty(){

return nextIndex<=0;

}

public void printQueue(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue(10);

q1.printQueue(); //[empty]

System.out.println("Size of the Queue : "+q1.size());//0

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

System.out.println("Size of the Queue : "+q1.size());//6

int data=q1.peek();

System.out.println("First element of the Queue : "+data);//10

q1.printQueue(); //[10, 20, 30, 40, 50]

data=q1.poll();

System.out.println("First element of the Queue : "+data); //10

q1.printQueue(); //[20, 30, 40, 50]

data=q1.peek();

System.out.println("First element of the Queue : "+data);//20

q1.printQueue(); //[20, 30, 40, 50]

data=q1.poll();

System.out.println("First element of the Queue : "+data); //20

q1.printQueue(); //[30, 40, 50]

q1.clear();

q1.printQueue(); //[empty]

data=q1.peek();

System.out.println("First element of the Queue : "+data);//-1

data=q1.poll();

System.out.println("First element of the Queue : "+data); //-1

}

}

//////////////////////////////////////////////////////////////////////////////

Q93 From Step III

=================

class Queue{

private int nextIndex;

private int[] dataArray;

Queue(int size){

dataArray=new int[size];

nextIndex=0;

}

public void deQueue(){

if(isEmpty()){

System.out.println("Queue is empty..");

}else{

for(int i=0; i<nextIndex-1; i++){

dataArray[i]=dataArray[i+1];

}

nextIndex--;

}

}

private boolean isFull(){

return nextIndex>=dataArray.length;

}

private void createNewArray(){

int[] newArray=new int[2\*dataArray.length];

for(int i=0; i<nextIndex; i++){

newArray[i]=dataArray[i];

}

dataArray=newArray;

}

public void enQueue(int data){

if(isFull()){

createNewArray();

}else{

dataArray[nextIndex++]=data;

}

}

public int size(){

return nextIndex;

}

public int peek(){

return isEmpty()? -1 : dataArray[0];

}

public int poll(){

if(isEmpty()){

return -1;

}else{

int data=dataArray[0];

deQueue();

return data;

}

}

public void clear(){

nextIndex=0;

}

private boolean isEmpty(){

return nextIndex<=0;

}

public void printQueue(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue(10);

q1.printQueue(); //[empty]

System.out.println("Size of the Queue : "+q1.size());//0

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

System.out.println("Size of the Queue : "+q1.size());//5

int data=q1.peek();

System.out.println("First element of the Queue : "+data);//10

q1.printQueue(); //[10, 20, 30, 40, 50]

data=q1.poll();

System.out.println("First element of the Queue : "+data); //10

q1.printQueue(); //[20, 30, 40, 50]

data=q1.peek();

System.out.println("First element of the Queue : "+data);//20

q1.printQueue(); //[20, 30, 40, 50]

data=q1.poll();

System.out.println("First element of the Queue : "+data); //20

q1.printQueue(); //[30, 40, 50]

q1.clear();

q1.printQueue(); //[empty]

data=q1.peek();

System.out.println("First element of the Queue : "+data);//-1

data=q1.poll();

System.out.println("First element of the Queue : "+data); //-1

}

}

//////////////////////////////////////////////////////////////////////////////

Class List

//////////////////////////////////////////////////////////////////////////////

Q94

class List{

private int nextIndex;

private int[] dataArray;

}

class Demo{

public static void main(String args[]){

List list=new List(10);

list.add(10);

list.add(20);

list.add(40);

list.add(50);

list.printList(); //[10,20,40,50]

//----------------------------------------

list.add(2,30);

list.printList(); //[10,20,30,40,50]

list.add(2,99);

list.printList(); //[10,20,99,30,40,50]

list.remove(2);

list.printList(); //[10,20,30,40,50]

//----------------------------------------

list.add(new int[]{60,70,80,90});

list.printList(); //[10,20,30,40,50,60,70,80,90]

System.out.println("Size : "+list.size()); //9

//---------------------------------------

list.add(3,new int[]{1,2,3,4,5});

list.printList(); //[10,20,30,1,2,3,4,5,40,50,60,70,80,90]

System.out.println("Size : "+list.size()); //14

//--------------------------------------

list.remove(3,7);

list.printList(); //[10,20,30,40,50,60,70,80,90]

System.out.println("Size : "+list.size()); //9

//---------------------------------------

int[] items=list.toArray();

for(int a : items){

System.out.print(a+" "); //10 20 30 40 50 60 70 80 90

}

System.out.println();

//---------------------------------------

int[] item2=list.toArray(4, 7);

for(int a : items2){

System.out.print(a+" "); //50 60 70 80

}

System.out.println();

}

}

//////////////////////////////////////////////////////////////////////////////

Dynamic Linked Stack

//////////////////////////////////////////////////////////////////////////////

Q95 Step I

==========

class Node{

int data;

Node next;

}

class Demo{

public static void main(String args[]){

Node t1;

t1=new Node();

}

}

//////////////////////////////////////////////////////////////////////////////

Q96 Step II

===========

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Demo{

public static void main(String args[]){

Node t1=new Node(100);

}

}

//////////////////////////////////////////////////////////////////////////////

Q97 Step III

============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Demo{

public static void main(String args[]){

Node t1=new Node(100);

t1.next=new Node(200);

}

}

//////////////////////////////////////////////////////////////////////////////

Q98 Step IV

===========

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Demo{

public static void main(String args[]){

Node t1=new Node(100);

t1.next=new Node(200);

t1.next.next=new Node(300);

t1.next.next.next=t1;

}

}

//////////////////////////////////////////////////////////////////////////////

Q99 Step V

==========

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Demo{

public static void main(String args[]){

Node t1=new Node(100);

Node t2=new Node(100);

Node t3=new Node(100);

Node top;

top=t1;

t1.next=t2;

t2.next=t3;

t3.next=t1;

}

}

//////////////////////////////////////////////////////////////////////////////

Q100 Step VI

============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

}

}

//////////////////////////////////////////////////////////////////////////////

Q101 Step VII

=============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

}

}

//////////////////////////////////////////////////////////////////////////////

Q102 Exercise

=============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

top=n1;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

}

}

//////////////////////////////////////////////////////////////////////////////

Q103 Step VIII

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.printStack(); //[40,30,20,10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q104 From Q103

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void printStack(){

System.out.print("[");

//

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.printStack(); //[40,30,20,10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q105 From Q104

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.printStack(); //[40,30,20,10]

s1.printStack(); //[40,30,20,10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q106 Step IX

============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.printStack(); //[40,30,20,10]

s1.pop();

s1.printStack(); //[30,20,10]

s1.pop();

s1.printStack(); //[20,10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q107 From Q106

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.printStack(); //[40,30,20,10]

s1.pop();

s1.printStack(); //[30,20,10]

s1.pop();

s1.printStack(); //[20,10]

}

}

//////////////////////////////////////////////////////////////////////////////

Q108 Step X

===========

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.printStack(); //[empty]

System.out.println("Size of the stack : "+s1.size());

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

s1.pop();

s1.printStack(); //[40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//4

s1.clear();

s1.printStack(); //[empty]

System.out.println("Size of the stack : "+s1.size());//0

}

}

//////////////////////////////////////////////////////////////////////////////

Q109 From Q108

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.printStack(); //[empty]

System.out.println("Size of the stack : "+s1.size());

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

s1.pop();

s1.printStack(); //[40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//4

s1.clear();

s1.printStack(); //[empty]

System.out.println("Size of the stack : "+s1.size());//0

}

}

//////////////////////////////////////////////////////////////////////////////

Q110 Step XI

============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

int data;

data=s1.peek();

System.out.println("First element (called peek) of the stack : "+data);//50

s1.printStack(); //[50,40,30,20,10]

data=s1.poll();

System.out.println("First element (called poll) of the stack : "+data);//50

s1.printStack(); //[40,30,20,10]

s1.clear();

data=s1.peek();

System.out.println("First element (called peek) of the stack : "+data);//-1

s1.printStack(); //[empty]

data=s1.peek();

System.out.println("First element (called poll) of the stack : "+data);//-1

s1.printStack(); //[empty]

}

}

//////////////////////////////////////////////////////////////////////////////

Q111 From Q110

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

public int peek(){

return isEmpty() ? -1 : top.data;

}

public int poll(){

if(isEmpty()){

return -1;

}else{

int data=top.data;

top=top.next;

return data;

}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

int data;

data=s1.peek();

System.out.println("First element (called peek) of the stack : "+data);//50

s1.printStack(); //[50,40,30,20,10]

data=s1.poll();

System.out.println("First element (called poll) of the stack : "+data);//50

s1.printStack(); //[40,30,20,10]

s1.clear();

data=s1.peek();

System.out.println("First element (called peek) of the stack : "+data);//-1

s1.printStack(); //[empty]

data=s1.peek();

System.out.println("First element (called poll) of the stack : "+data);//-1

s1.printStack(); //[empty]

}

}

//////////////////////////////////////////////////////////////////////////////

Q112 Step XII

=============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

public int peek(){

return isEmpty() ? -1 : top.data;

}

public int poll(){

if(isEmpty()){

return -1;

}else{

int data=top.data;

top=top.next;

return data;

}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

int[] dataArray=s1.toArray();

for(int data : dataArray){

System.out.print(data+" ");

}

}

}

//////////////////////////////////////////////////////////////////////////////

Q113 From Q112

==============

class Node{

int data;

Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

public int peek(){

return isEmpty() ? -1 : top.data;

}

public int poll(){

if(isEmpty()){

return -1;

}else{

int data=top.data;

top=top.next;

return data;

}

}

public int[] toArray(){

int[] dataArray=new int[size()];

Node temp=top;

for(int i=0; i<size();i++){

dataArray[i]=temp.data;

temp=temp.next;

}

return dataArray;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

int[] dataArray=s1.toArray();

for(int data : dataArray){

System.out.print(data+" ");

}

}

}

//////////////////////////////////////////////////////////////////////////////

Q114 Exercise

=============

class Node{

private int data;

private Node next;

Node(int data){this.data=data;}

}

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

public int peek(){

return isEmpty() ? -1 : top.data;

}

public int poll(){

if(isEmpty()){

return -1;

}else{

int data=top.data;

top=top.next;

return data;

}

}

public int[] toArray(){

int[] dataArray=new int[size()];

Node temp=top;

for(int i=0; i<size();i++){

dataArray[i]=temp.data;

temp=temp.next;

}

return dataArray;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

int[] dataArray=s1.toArray();

for(int data : dataArray){

System.out.print(data+" ");

}

}

}

//////////////////////////////////////////////////////////////////////////////

Q115 From Q114 (Using Inner Classes)

====================================

class Stack{

private Node top;

public void push(int data){

Node n1=new Node(data);

n1.next=top;

top=n1;

}

private boolean isEmpty(){

return top==null;

}

public void clear(){

top=null;

}

public int size(){

Node temp=top;

int count=0;

while(temp!=null){

count++;

temp=temp.next;

}

return count;

}

public void printStack(){

System.out.print("[");

Node temp=top;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

public void pop(){

if(!isEmpty()){

top=top.next;

}

}

public int peek(){

return isEmpty() ? -1 : top.data;

}

public int poll(){

if(isEmpty()){

return -1;

}else{

int data=top.data;

top=top.next;

return data;

}

}

public int[] toArray(){

int[] dataArray=new int[size()];

Node temp=top;

for(int i=0; i<size();i++){

dataArray[i]=temp.data;

temp=temp.next;

}

return dataArray;

}

class Node{

private int data;

private Node next;

Node(int data){this.data=data;}

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack();

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50,40,30,20,10]

System.out.println("Size of the stack : "+s1.size());//5

int[] dataArray=s1.toArray();

for(int data : dataArray){

System.out.print(data+" ");

}

}

}

//////////////////////////////////////////////////////////////////////////////

Class "Queue"

//////////////////////////////////////////////////////////////////////////////

Q116 Step I

===========

class Queue{

private Node front;

class Node{

private int data;

private Node next;

Node(int data){this.data=data;}

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue();

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

}

}

//////////////////////////////////////////////////////////////////////////////

Q117 From Q116

==============

class Queue{

private Node front;

class Node{

private int data;

private Node next;

Node(int data){this.data=data;}

}

public void enQueue(int data){

Node n1=new Node(data);

if(front==null){

front=n1;

}else{

Node temp=front;

while(temp.next!=null){

temp=temp.next;

}

temp.next=n1;

}

}

private boolean isEmpty(){

return front==null;

}

public void printQueue(){

System.out.print("[");

Node temp=front;

while(temp!=null){

System.out.print(temp.data+", ");

temp=temp.next;

}

System.out.println(isEmpty() ? "empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Queue q1=new Queue();

q1.enQueue(10);

q1.enQueue(20);

q1.enQueue(30);

q1.enQueue(40);

q1.enQueue(50);

q1.printQueue(); //[10, 20, 30, 40, 50]

}

}