1. Write the class Date having attributes like day, month & year. Add default & parameterized constructors. Add getters & setters. Add method to print the date. Add method to swap two dates.

**package** zensar;

**public** **class** Day11 {

Day11()

{

**this**.day="sunday";

**this**.month="january";

**this**.year="2021";

}

Day11(String day,String month,String year)

{

**this**.day=day;

**this**.month=month;

**this**.year=year;

}

**private** String day;

**private** String month;

**private** String year;

**public** **void** setDay(String day)

{

**this**.day=day;

}

**public** **void** setmonth(String month)

{

**this**.month=month;

}

**public** **void** setyear(String year)

{

**this**.year=year;

}

**public** String getDay()

{

**return** day;

}

**public** String getMonth()

{

**return** month;

}

**public** String getyear()

{

**return** year;

}

**public** **static** **void** printDate()

{

Day11 day11 = **new** Day11();

String day=day11.getDay();

String month=day11.getMonth();

String year=day11.getyear();

System.***out***.println("stored date : " + day+ "/" + month+"/" + year);

}

**public** **static** **void** swapDate()

{

Day11 day11 = **new** Day11();

String day=day11.getDay();

String month=day11.getMonth();

String year=day11.getyear();

day11.setDay("thursday");

String day2=day11.getDay();

System.***out***.println(" before swapping = "+ day + " " + day2);

System.***out***.println("stored date : " + day+ "/" + month+"/" + year);

String temp;

temp=day;

day=day2;

day2=temp;

System.***out***.println(" after swapping ="+ day + " " + day2);

System.***out***.println("stored date : " + day+ "/" + month+"/" + year);

}

**public** **static** **void** main(String[] args) {

*printDate*();

*swapDate*();

}

}

Output:

stored date : sunday/january/2021

before swapping = sunday thursday

stored date : sunday/january/2021

after swapping =thursday sunday

stored date : thursday/january/2021

2.Write a class ComplexNumber having attributes real & imaginary. Add functions like add, subtract, multiply & swap.

**package** zensar;

**public** **class** Complex {

// **TODO** Auto-generated method stub

**double** real, img;

//constructor to initialize the complex number

Complex(**double** r, **double** i){

**this**.real = r;

**this**.img = i;

}

**public** **static** Complex sum(Complex c1, Complex c2)

{

//creating a temporary complex number to hold the sum of two numbers

Complex temp = **new** Complex(0,0);

temp.real = c1.real + c2.real;

temp.img = c1.img + c2.img;

//returning the output complex number

**return** temp;

}

**public** **static** **void** main(String[] args) {

Complex c1 = **new** Complex(5.5, 4);

Complex c2 = **new** Complex(1.2, 3.5);

Complex temp = *sum*(c1, c2);

System.***out***.println("Sum is: "+ temp.real+" + "+temp.img+"i");

}

}

Output:

Sum is: 6.7 + 7.5i

3.Write a class Account & add methods like deposit, withdraw, print etc.

**package** zensar;

**public** **class** Account {

**int** account\_no;

**float** balance;

**public** Account(**int** acc,**float** bal)

{

account\_no=acc;

balance=bal;

}

**public** **void** withdraw(**int** w)

{

balance=balance-w;

}

**public** **void** deposit(**int** d)

{

balance=balance+d;

}

**public** **void** display()

{

System.***out***.println("account number is" + account\_no);

System.***out***.println("balance is" + balance);

}

**public** **static** **void** main(String[] args) {

Account a =**new** Account(3422,1000);

a.withdraw(500);

a.deposit(700);

a.display();

}

}

Output:

account number is3422

balance is1200.0

1. Write a program to implement a Stack using arrays as follows-

class StackedArray {

int ary[];

push(--) { }

pop() {--) {}

}

**package** zensar;

**public** **class** StackedArray {

**int** arry[]=**new** **int**[5];

**int** top=0;

**public** **void** push(**int** data)

{

arry[top]=data;

top++; }

**public** **void** pop()

{

**int** data;

top--;

arry[top]=0;

}

**public** **void** show()

{

System.***out***.println("array elements are : " );

**for**(**int** i=0;i<arry.length;i++)

{

System.***out***.println("arr[" + i +"] ="+arry[i]);

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

StackedArray stackedarray = **new** StackedArray();

stackedarray.push(1);

stackedarray.push(2);

stackedarray.push(3);

stackedarray.push(4);

stackedarray.push(5);

stackedarray.show();

stackedarray.pop();

stackedarray.show();

stackedarray.pop();

stackedarray.show();

}

}

1. Write a program to implement a Queue using arrays as follows-

class QueuedArray {

int ary[];

push(--) { }

pop() {--) {}

}

**package** zensar;

**public** **class** QueuedArray {

**int** size=5;

**int** [] arry=**new** **int**[size];

**int** rear=-1;

**int** front=-1;

**public** **void** enqueue(**int** data)

{

**if**(rear==size-1)

{

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

}

**if**(front==-1 && rear == -1) {

front++;

rear++;

arry[rear]=data;

}

**else**

rear++;

arry[rear]=data;

}

**public** **void** show()

{

**for**(**int** i=0;i<arry.length;i++)

{

System.***out***.println("arry["+ i + "] = " +arry[i]);

}

}

**public** **void** deque()

{

**if**(front==-1 && rear == -1)

{

System.***out***.println("array is empty");

}

**else** **if**(front==rear)

{

arry[front]=0;

front = rear =-1;

}

**else**

{

**for**(**int** i=0;i<=rear;i++)

{

arry[i]=arry[i+1];

}

--rear;

}

}

**public** **static** **void** main(String[] args) {

QueuedArray queue=**new** QueuedArray();

queue.enqueue(11);

queue.enqueue(15);

queue.enqueue(20);

queue.enqueue(25);

System.***out***.println("queue elements are :");

queue.show();

System.***out***.println("removing front element from queue");

queue.deque();

queue.show();

}

}

1. Write a single tone class. Confirm that single tone class cannot be inherited.

**public** **class** Singletone {

**private** **static** Singletone *singletone*=**null**;

**public** **static** Singletone singletoneMethod()

{

**if**(*singletone* == **null**)

{

System.***out***.println("object is null---create object");

*singletone*=**new** Singletone();

**return** *singletone*;

}

**else**

{

System.***out***.println("object cannot be created");

**return** *singletone*;

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Singletone singletone=Singletone.*singletoneMethod*();

Singletone singletone1=Singletone.*singletoneMethod*();

}

}

Output:

queue elements are :

arry[0] = 11

arry[1] = 15

arry[2] = 20

arry[3] = 25

arry[4] = 0

removing front element from queue

arry[0] = 15

arry[1] = 20

arry[2] = 25

arry[3] = 0

arry[4]=0

7.Write java classes to build doubly linked list. Add functionalities like add new node, insert node, delete node, count nodes & print linked list.

class Node {

Node previous;

Node next;

Int data;

class Node {

Node previous;

Node next;

Int data;

}

**package** com.zensar;

**public** **class** Doublylinkedlist

{

Node head,tail = **null**;

**class** Node

{

**int** data;

Node prev;

Node next;

Node(**int** d)

{

data = d;

}

}

**void** insert(**int** data)

{

Node new\_node = **new** Node(data);

**if**(head==**null**)

{

head = tail = new\_node;

head.prev = **null**;

tail.next = **null**;

}

tail.next = new\_node;

new\_node.prev = tail;

tail = new\_node;

new\_node.next = **null**;

}

**void** delete(Node del)

{

**if**(head == **null** )

{

**return**;

}

**if**(head == del) {

head = del.next;

}

**if**(del.next != **null**) {

del.next.prev = del.prev;

}

**if**(del.prev != **null**) {

del.prev.next = del.next;

}

**return**;

}

**void** printNodes()

{

Node curr = head;

**if**(head == **null**)

{

System.***out***.println("DLL is empty");

**return**;

}

**while**(curr!=**null**)

{

System.***out***.print(curr.data + "->" );

curr = curr.next;

}

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

}

**public** **int** countNodes() {

**int** counter = 0;

Node current = head;

**while**(current != **null**) {

counter++;

current = current.next;

}

**return** counter;

}

**public** **static** **void** main(String[] args) {

Doublylinkedlist dl = **new** Doublylinkedlist();

dl.insert(35);

dl.insert(38);

dl.insert(45);

dl.insert(55);

dl.printNodes();

dl.delete(dl.head.next);

dl.delete(dl.tail.prev);

System.***out***.println("after deletion");

dl.printNodes();

System.***out***.println("No. of nodes: "+ dl.countNodes());

}

}

Output:

