Q 1 :

package com.company;

import java.util.\*;

abstract class Parent{

abstract public void message();

}

class subclass1 extends Parent{

@Override

public void message() {

System.out.println("This is first subclass");

}

}

class subclass2 extends Parent{

@Override

public void message() {

System.out.println("This is second subclass");

}

}

public class Main {

public static void main(String[] args) {

subclass1 a1 = new subclass1();

subclass2 a2 =new subclass2();

a1.message();

a2.message();

}}

Q 2 :

package com.company;

import java.util.\*;

public class Main {

public static void main(String[] args) {

bank\_A a1 =new bank\_A();

bank\_B a2 = new bank\_B();

bank\_C a3 = new bank\_C();

a1.get\_balance();

a2.get\_balance();

a3.get\_balance();

}

}

abstract class bank{

abstract public void get\_balance();

}

class bank\_A extends bank{

@Override

public void get\_balance() {

System.out.println("100$");

}

}

class bank\_B extends bank{

@Override

public void get\_balance() {

System.out.println("150$");

}

}

class bank\_C extends bank{

@Override

public void get\_balance() {

System.out.println("200$");

}}

Q 5 :

package com.company;

import java.util.\*;

public class Main {

public static void main(String[] args) {

Cats c1 = new Cats();

Dogs d1 = new Dogs();

c1.cats();

d1.dogs();

}

}

abstract class animals{

abstract public void cats();

abstract public void dogs();

}

class Cats extends animals{

@Override

public void cats() {

System.out.println("Cats meow");

}

@Override

public void dogs() {}

}

class Dogs extends animals{

@Override

public void dogs(){

System.out.println("Dogs Bark");

}

@Override

public void cats() {}

}

Q 7 :

ackage com.company;

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner t=new Scanner(System.in);

System.out.println("Enter size of array; ");

int s=t.nextInt();

int m[] =new int[s];

for(int x=0;x<s;x++) {

System.out.println("Enter "+(x+1)+" element: ");

m[x]=t.nextInt();

}

Array c1 = new Array();

c1.add(m);

c1.remove(m);

c1.check(m);

}

}

interface queue{

void add(int m[]);

void remove(int m[]);

void check(int m[]);

}

class Array implements queue{

Array(){}

@Override

public void add(int m[]) {

int tot=0;

for(int x=0;x<m.length;x++) {

tot=tot+m[x];

}

System.out.println("sum is "+tot);

}

@Override

public void remove(int m[]) {

Scanner t=new Scanner(System.in);

System.out.println("Enter the element to Remove: ");

int r=t.nextInt();

for(int x=0;x<m.length;x++)

if(m[x]==r) {

int tem=m[x];

m[x]=m[m.length-1];

m[m.length-1]=m[x];

m[x]=tem;

}

System.out.print("Element is removed: ");

for(int x=0;x<m.length-1;x++) {

System.out.print(m[x]+" ");

}}

@Override

public void check(int m[]) {

if(m==null)

System.out.println("Array is empty:");

else

System.out.println("Array isn't empty: ");

}}