Test_Interface_1

```
using System;
using System.Collections.Generic.
Using System.Text.
namespace Test_Interface_1{
interface BasicCalculatorInterface{
int sum(int x, int y);
int sub(int x, int y);
int multiplication(int x, int y);
int division (int x, int y) }}

using System;
using System.Collection.Generic;
using System.Text;
namespace Test_Interface_1
class ScientificCalculator Interface : BasicCalculator
Interface{
int add;
int subt;
int mult;
    int div;
double pow;
```

```
publie int sum(int x, int y){
return add =x+y; }
Publie int sub(int x, inty){
return sub =x-y;}
public int multiplication(int x, int y){
return mult=x*y;}
publie int division(int x, int y){
return dir= x/y;}
publie double toThePow(int x, inty){
return pow= Math.Pow(x,y);}
public void show(){
Console.Writeline("Summation Result:"+add);
Console.Writeline("Subtract Result:"+subt);
Console.WriteLine("Division Result:" + div);
Console.WriteLine("Math.Pow() Result:"+Pow);
Console.WriteLine("Multiplication Result:"+mult);
}
}}
```

```
Using System;
namespace Test_Interface_1{
class Program{
Static void Main(string[] args){
SeientificCalculatorInterface se = new
SelentificCalculatorInterface();

 se.sum(5,5);
se.sub(5,5);
se.multiplication(5,5);
se.division(5,5);
se.toThePow(5,5);
se.show();

Console.ReadLine();
}
  }
    }
```

(15)

Test_Interface_2

```
using System;
using System.Collection.Generic;
using System.Text;

namespace Test_Interface_2{
interface BasicBankingInterface{
 bool deposit(double amount);
 bool withdraw(double amount);
}
}

using System;
using System.Collections.Generic;
using System.Text;

namespace Test_Interface_2{
class Account : BasicBankingInterface{
double balance;
string accType;
string id;
double saving_balance;
double OVERDRAFT_LIMIT = 5000;

public Account(){}
```

```
public Account(string accType, string id, double balance)
{ this.accType = accType;
  this.id = id;
  this.balance = balance; }
public string Type{
get{return accType;}
set{accType = value;}}

public string Id{
get{return id;}
set{ id = value;}}

public double Balance{
get{return balance;}
set{ balance = value;}

}

public bool deposit (double amount){
if (amount > 0){
```

```
balance = balance + amount;
Console.WriteLine("\n After Deposit your Account
        Balance :" + balance + "\n");
return true; }
else{
Console.WriteLine("Amount is very low can not Deposit");
return false;
}
public bool withdraw(double amount)
if(balance >= amount && "Cur" == accType.SubString(0,3))
{balance - amount;
Console.Write("\n-----Current Account ---- \n");
Console.WriteLine("-->withdraw Balance :" + amoun);
Console.WriteLine("-->After withdraw your Account
        balance :" + balance + "\n");
return true;
}
else if(balance >= amount && "Sav" == accType.Substring(0,3))
{
saving_balance = 0.8* balance;
balance = balance - saving_balance;
```

```
Console.WriteLine("\n--> Avaible Balance :"+balance+"\n");
if (balance>=amount && aceType == "Saving"){
balance = balance-amount;
Console.WriteLine("\n------Saving Accoun-------\n");
Console.WriteLine("--> withdrawed Balance :"+amount);
Consol e.WriteLine("--> After withdraw your Accoant
                    balance :" + balance+"\n");
Console.WriteLine("--> your saving balance :" +
                    saving_balance + "\n");
balance = saving_balance + balance;
return true;}

else{
Console.WriteLine("---Cannot withdraw Saving Balance-
                    --");
balance=saving_balance + balance;
return false;
}
}
```

```
else if(balanee==0 && "Ove"==accType.Substring(0,3))
{
 if(balanee<=0 && OVERDRAFT_LIMIT!=0){
balanee=OVERDRAFT_LIMIT;
OVERDRAFT_LIMIT=OVERDRAFT_LIMIT - balanee;
Console.WriteLine("\n--- Overdraft Suesstull---\n");
Console.WriteLine("Now your Account balanee:"
                  + balanee + "\n");
return true;}
else{
Console.WriteLine("---OverDraft Unsuesstull---");
return false;}

else{
Console.WriteLine("Cannot Withdraw Please
        eReek your Balanee---");
return false;
}
}
```

```
public void change_acc_type(string type)
{
Console.WriteLine("--> change Account type
    [" + this.accType + "] to[" + type + "]");
  this.acctype = type;
}
}
}
using System;
using System.Collections.Generic;
using System.Text;
namespace Test_Interface_2{
class current : Account{
double balance;
string accType;
string id;
public current(){}
public current(string accType, string id, double balance)
{
this.accType = acc_Type;
this.id = id;
this.balance = balance; }
```

```
publie string Type{
get{return accType;}
set{ accType =value;}}

publie string Id{
get{return id;}
set{ id= value; }}

publie doubleBalanee{
{get{return balanee;}
  set{balanee =value;}}

publie bool deposit(doubleamount){
if(amount>0){
balanee =balanee+amount;
console.WriteLine("After Deposit your Account
 Balanee :"+balanee+"\n")
returntrue;}

else{
Console.WriteLine("Amount is very low can not
                   Deposit");
return false;
}}
```

```
public bool withdraw (double amount) {
if(balanee>amount && "eur" == accType.Substring(0,3))
{
 balanee = balanee - amount;
Console.WriteLine("withdraw Balanee :" + amount);
Console.WriteLine(" After withdraw your Accounts
          balanee :" + balanee + "\n");
return true; }
else
{
Console.WriteLine("--- Cannot withdraw Please
check your Balanee ---");
return false;
}
}
public void change_acc_type (string type)
{
 this.accType = type;
}
}
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

name Test_Interface_2{

class Savings: Account{
double balance;
string accType;
string id;
double saving_balance;
public Savings() { }

public Savings(string accType, string id, double
                balance)

{
 this.accType = accType;
 this.id = id;
 this.balance = balance;
}
public string Type{
get{return accType}
set{accType = value}}
```

```
publie string Id{
get{return id;}
set{ id=value:}}

publie double balanee{
 get{return balanee;}
 set{ balanee=value;}}

public bool deposit (double amount){
if(amount>0){
 balance = balance + amount;
Consde.WriLine("After Deposit your Accoun
        Balanee;" + balanee + "\n");
 return true;}
 else{
Console.WriteLine("Amount is very low can not
              Deposit");
return false;}
}

public bool withdraw(double amount){
```

```
saving_balance = 0.8 * balance;
balance = balance - saving_balance;
if(balance >= amount && accType == "Saving") {
balance = balance - amount;
Console.WriteLine("Withdrawed Balance :" + amount);
Console.WriteLine(" your saving balance : + saving_balan
                  ce + "\n");
return true; }
else {
Console.WriteLine("--- cannot withdraw please
                  check your Balance---");
return false; } }

public void change_acc_type(String type)
{
 this.accType = type;
}
}
}
```

```
using sytem;
namespace Test_Interface_2
{class Program{
static void main(string[] args){
 Account cur=new Current();
 cur.type="Current";
cur.deposit(100);
cur.withdraw(10);
cur.change_acc_type("Saving");
cur.withdraw(10);

Account sav=new Saving();
sav.type="Saving";
sav.deposit(100);
sav.withdraw(20);
sav.change_acc_type("Current");
sav.withdraw(80);


Account ove =new Overdratt();
Ove.Type="Overdratt";
Ove.withdraw(500);
ove.chang_acc_type("Saving");
```

```
using system;
using system.Collection.Generic;
Using System.Text;
namespace Test-Interface-2{
class Overdraft : Account{
double balance;
String accType;
string id;
double OVERDRAFT_LIMIT =5000;
bool od;
public Overdraft(){}
public overdraft (string accType, string id, double
                  balance){
this.accType = accType;
this.id =id;
this.balance = balance;}

public string Type{
get{return accType;}
set{accType =value;}}

public string Id{
get{return id;}
set{balance =value;}}
```

```
public bool withdraw(double amount){
if(balanee==0)
{
balanee =balanee-amount;

Console.WriteLine("Withdraw Balanee:"+amount);

Console.WriteLine("After withdraw your Account

Balanee.:" + balanee +"\n");

return true;
}
else{
Console.WriteLine("---Cannot withdraw please
                    check your balanee");
return false;}}

public void change_acc-type(string type)
{
 this.accType=type;
}
}
}
```

Test_Interface_3

```
using System;
using System.Collection.Generic;
using System.Text;

namespace Test_Interface_3{
interface RadioPlayerInterface{
void swit(bool on);
void retune(double frequency);
void setVolume(int loudness);
void changeChannel();
  }
}
```

```
using System;
using System.Collection.Generic;
using System.Text;

namespace Test_Interface_3
{
```

```
class MusicPlayerInterface:RadioPlayerInterface{
bool swite;
bool plays;
bool next;
bool prev;
MusicFil[] listofMusic =new MusicFil[500];
int totalMusic =1;
int i=0;
public void swit (bool on){
  swite =on;
  if(swite){
  Console.WriteLine("switeR on"); }
  else{
  Console.WriteLine("switeR of");}
}
public void retune(double frequency){
  Console.WriteLine("Frequeney :"+frequeney+"HZ");
}
public void setVolume(int loudness){
Console.WriteLine("Loudness ;"+loudness);
}
```

(6)

```
public void eRangeeRannel(){
Console.WriteLine("eRannel eRange");}

public void play (bool on){
plays=on;
if(plays){
Console.WriteLine("\n-----Play Musie------");

Console.WriteLine("Music Tittle : "+listofMusic[i].Title);
Console.WriteLine("Music Artist: "+listofMusic[i].Artist);
Console.WriteLine("Music year of Release ; +
                    listofMusic[i].Release_year);
Console.WriteLine("Music Duration of Seconds :" +
                    listofMusic[i].Duration);
Console.WriteLine("Music Serial Number :"+i+"\n");
}
else{
Console.WriteLine("Not Responding");

}

}
```

(7)

```
public void playNext (bool m_next){
next = m_next;
if(next){
Console.WriteLine("\n----- Play next Music- ----");
i++
Console.WriteLine("Music Title : "+listofMusic[i].Title);
Console.WriteLine("Music Artist:" + list of Music[i].Artist);
Console.WriteLine("Music year of Release :" +
                  listofMusic[i].Release-year);
Console.WriteLine("Music Duration of seconds;"+
                  tisofMusic[i].Duration);
Console.WriteLine("Music Serial Number;"+i+
                  "\n");
}
else
{
Console.WriteLine("Not Responding");
}
}
```

```
publie void playPrevious(bool m-prev){
prev = m-prev;
if(prev)
{
  --i;
Console.WriteLine("\n------Play previous Musie-----");
Console.WriteLine("Musie Title;"+listof Musie[i].Title);
Console.WriteLine("Musie Artist:"+ listofMusie[i].Artist);
Console.Writeline("Musie year of Release:"+
                  listofMusie[i].Release-year);
Console.WriteLine("Musie Duration of Seconds:")
                  +listofMusie[i].Duration);

Console.WriteLine("Musie Serial Number:"+it
                  "\n");
}
else
{
Console.WriteLine("Not Responding");

}

}
```

```
publie void AddnewMusie (MusicFill musie){
if (totalMusie < 500){
listofMusic[totalMusic]= music;
Console.WriteLine("\n--> Added New Music :" +
        listofMusic[total Music].Title);
Console.Writeline("--> Total Music :"+ totalMusic);
totalMusie++;
}
else
{
Console.WriteLine("--> canNot Added New Musie
                                    \n");
}
}
}
}
```

```
using System;
using System.Collection Generic;
using System.Text;
namespaee Test-Interface-2{
class MusicFil{
string title;
string artist;
int yearofRelease;
int durationInseconds;
public MusicFil(){}
public MusicFil(string title, string artist, int
  yearofRelease, int durationInseconds){
this.title=title;
this.artist= artist;
this.yearofRelease = yearof Release;
this.durationInseconds = durationInSeconds;
}
```

```
publie String Title
{
 set{title= value;}
 get{return title;}
}
publie string Artist{
set{artist =value;}
get{return artist;}
}
public int Release_Year{
 set{yearofRelease= value;}
 get{ return yearofRelease;}
}
public int Duration{
set{durationInSeconds=value;}
get{return durationInSeconds}
}
public void eRangeTitle (string title){
this.title=title;
}
```

```
public void show(){
Console.WriteLine("Music Title :" + title);
Console.WriteLine("Music Artist :" + artist);
ConsoleWriteLine("Music Year of Release :" +
                 yearofRelease);
Console.Writeline("Music Duration of Seconds:"+
                  durationInSeconds);
}
}
}
```

```
using System
namespace Test_Interface_3{
class Program{
static void Main(string[] args){
MusicPlayer mi= new MusicPlayerInterface();
MusicFilmt1=new MusicFil("AA","MM",2021,180);
//mt1.show();
//mt1. changeTitle("BB");
// mt.show();

MusicFil mt2=newMusicFil("CC","NN",2021,120);

MusicFil mt3 =new MusicFil("DD","OO",2021, 90);

MusicFil mt4 =new MusicFil("EE","PP", 2021, 110);

MusicFil mt4=new MusicFil("FF","WW",2021,80);


 mi.swit(true);

 mi.retune(50);

mi.setVolume(10);

mi.changeChannel();

mi.AddNewMusic(mt1);
```

```
mi.AddNewMusic(mt2);
mi.AddNewMusic(mt3);
mi.AdNewMusic(mt4);
mi.AddNewMusic(mt5);
mi.play(true);
mi.playNext(true);
mi.playPrevious(true);

Console.ReadLine();
}
}
}
```