OBJECT ORIENTED PROGRAMMING (CT-260)

ASSIGNMENT:01

TAQI HAIDER_CSIT_SECTION:B_ROLL#92

Exercise:-
Q1:-

```
#include<iostream>
#include <cmath>
using namespace std;
class num{
    double number;
    public:
        num(): number(0){}
        num(double n){number = n;}
        void getnum(){
            double n;
            cout<<"Enter number: "<<endl;
            cin>>n;
            number = n;
        }
        void shownum(){
            cout<<number<<endl;
        }
        void getratio(num obj){
            double A,B,ratio1=0,ratio2=0;
            if(number>obj.number){
                A=number;B=obj.number;
            }
            else{
                A=obj.number;B=number;
            }
            ratio1=(A+B)/A;
            ratio1=dot3place(ratio1);
            ratio2=A/B;
            ratio2=dot3place(ratio2);
            if(ratio1==ratio2){
                cout<< A << " and " << B << " Are in Golden Ratio"<<endl;
            }
            else{
                cout<< A << " and " << B << " Are NOT in Golden Ratio"<<endl;
            }
        }
        double dot3place(double ratio){
```

```
            ratio=ratio*1000;
            ratio=round(ratio);
            ratio=ratio/1000;
            return ratio;
        }
};
int main(){
    num n1,n2;
    n1.getnum();
    n2.getnum();
    n1.getratio(n2);
}
```

Q2:-

```
#include<iostream>
#include<cmath>
using namespace std;

class order{
    int DineIn=0,L_D_MN=0,Happyhour=0,cash=1;   //variables to store what option
the user wants, L_D_MN is Lunch_Dinner_MidNight choice.
    double Bill,discount=0,discamount=0;
    public:
        void getdata(){

            char selection; //declaration to use char inputs later.
            cout<<"Please Enter the cost of your bill:"<<endl;
            cin>>Bill;

            cout<<"\nWhen would you like to take your meal?\n\n(A) Lunch\t(B)
Dinner\t(C) Midnight"<<endl;
            cin>>selection;
            while((selection!='a')&&(selection!='b')&&(selection!='c')&&(selectio
n!='A')&&(selection!='B')&&(selection!='C')){ //loops until correct input is
given
                cout<<"\nPlease Answer with the mentioned
Characters..."<<endl<<"When would you like to take your meal?\n\n(A) Lunch\t(B)
Dinner\t(C) Midnight"<<endl;
                cin>>selection;
            }
            if((selection=='a')||(selection=='A')){
                L_D_MN=1;
            }
            else if((selection=='b')||(selection=='B')){
                L_D_MN=2;
            }
```

```
            else if((selection=='c')||(selection=='C')){
                L_D_MN=3;
            }

            cout<<"\nWill you have this meal as:\n(A) Takeout\t(B) Dine-
In"<<endl;
            cin>>selection;
            while((selection!='a')&&(selection!='b')&&(selection!='A')&&(selectio
n!='B')){
                cout<<"\nPlease Answer with the mentioned Characters...\nWill you
have this meal as:\n(A) Takeout\t(B) Dine-In"<<endl;
                cin>>selection;
            }
            if((selection=='a')||(selection=='A')){
                DineIn=0;
            }
            else if((selection=='b')||(selection=='B')){
                DineIn=1;
            }

            cout<<"\nWould you like to avail Special Happy Hour deal instead of
the regular sale discount?\nA 50% Discount of up to RS.2500/- on your bill will
be applied:\n(Y) YES\t(N) NO"<<endl;
            cin>>selection;
            while((selection!='Y')&&(selection!='N')&&(selection!='y')&&(selectio
n!='n')){
                cout<<"\nPlease Answer with the mentioned Characters..."<<endl;
                cout<<"Would you like to avail Special Happy Hour deal instead of
the regular sale discount?\nA 50% Discount of up to RS.2500/- on your bill will
be applied:\n(Y) YES\t(N) NO"<<endl;
                cin>>selection;
            }
            if((selection=='y')||(selection=='Y')){ //If happy hour deal is
accepted, overwrites previously calculated discount with happy hour deal.
                discount=0.5;
                Happyhour=1;
            }
            cout<<"\nHow would you like to pay for your deal? (Cards will not get
Discount!)?\n\n(A) Credit Card\t(B) Debit Card\t(C) Cash"<<endl;
            cin>>selection;
            while((selection!='a')&&(selection!='b')&&(selection!='c')&&(selectio
n!='A')&&(selection!='B')&&(selection!='C')){
                cout<<"\nPlease Answer with the mentioned Characters..."<<endl;
                cout<<"How would you like to pay for your deal? (Cards will not
get Discount!)?\n\n(A) Credit Card\t(B) Debit Card\t(C) Cash"<<endl;
                cin>>selection;
            }
            if((selection=='A')||(selection=='a')||(selection=='B')||(selection==
'b')){
```

```
                cash=0;
                discount=0;
            }
        }

        void calcdiscount(){
            if(Happyhour==0&&cash==1){
                if((L_D_MN==1)&&(DineIn==1)){   //Lunch and Dine In
                        discount=0.25;
                }
                else if((L_D_MN==1)&&(DineIn==0)){ //Lunch and TakeAway
                    discount=0.35;
                }
                else if((L_D_MN==2)&&(DineIn==1)){ //Dinner and Dine In
                    discount=0.2;
                }
                else if((L_D_MN==2)&&(DineIn==0)){ //Dinner and TakeAway
                    discount=0.25;
                }
                else if((L_D_MN==3)&&(DineIn==1)){ //Midnight and Dine In
                    discount=0;
                }
                else if((L_D_MN==3)&&(DineIn==0)){ //Midnight and TakeAway
                    discount=0.15;
                }
            }

            discamount=Bill*discount;

            if((Happyhour==1)&&(discamount>2500)){ //This Discount Amount limit
only applies to Happy Hour Deal thus it is checked by Happy Hour == 1
                discamount=2500;
            }

            Bill=Bill-discamount;

            cout<<"\nYour Meal will now cost : "<<Bill<<" as a discount of
"<<discount*100<<" was applied according to your selection"<<endl;
        }
};

int main(){
    order Customer;
    Customer.getdata();
    Customer.calcdiscount();
}
```

Q3:-

```
#include<iostream>
#include<cmath>
using namespace std;

class encrypt{
    int message,encode;
    public:
        encrypt():message(0),encode(0){}
        encrypt(int m){message=m;}
        void get(){
            cout<<"Enter message to be encoded: "<<endl;
            cin>>message;
        }
        void show(){
            cout<<"Message : "<<message<<endl<<"Encoded : "<<encode<<endl;
        }
        void encoder(){
            int checks[10]={0,0,0,0,0,0,0,0,0,0}; //Array to count number of
occurences of a digit, i.e. checks[1]==0 means '1' appeared 0 times
            int count=0;
            count=(message==0)?1:log10(message)+1;
            int rem,ld,highest=-1,lowest=10;
            if(count<2){ //Checks if number was a single digit input, as
encodinging it wont give unique two digits output from input.
                cout<<"Number too small to actually encode as highest lowest will
be same after encoding"<<endl;
                return;
            }
            else if(message<0){ //converts num from negative to positive if
someone inputs negative number. Even though negative numbers dont mess up the
encoding, it is just in case if another compiler does not interact the same way.
                cout<<"Negative will be converted to positive to encode"<<endl;
                message = -message;
            }
            rem=message;
            for(int i=0;i<count;i++){ //looping through the whole number
                ld=rem%10;
                if(ld==0){
                    cout<<"Input Does NOT follow question's rule of being only in
1-9 range"<<endl; // this gives error if it encounters a zero in the number as
tasked by question's rule.
                    return;
                }
                rem=rem/10;
                if(ld>highest){highest=ld;}   //comparing with highest and then
with lowest
```

```
                if(ld<lowest){lowest=ld;}

                checks[ld]++;   //increments array element of index [Last Digit]
to later check for duplicates.

                if(checks[ld]>1){
                cout<<"Duplicate Numbers are present in Input value.\nQuestion's
rule of numbers appearing once at most was broken."<<endl; // this gives error if
any number was used more than once
                return;
                }
            }
            (count%2==0)?encode=(highest*10)+lowest:encode=(lowest*10)+highest;
        }

};
int main(){
    encrypt number;
    number.get();
    number.encoder();
    number.show();
}
```

Q4:-

```
 #include<iostream>
#include <stdlib.h>
using namespace std;

class time{
    int sec,min,hrs;
    public:
        time():sec(0),min(0),hrs(0){}
        time(int h,int m,int s){
            while(s>=60){
                s-=60;
                m++;
            }
            while(m>=60){
                m-=60;
                h++;
            }
            while(h>=12){
                h-=12;
            }
            hrs=h;min=m;sec=s;
        }
        time difference(time obj1){
```

```
            int A=(hrs*3600)+(min*60)+sec;int
B=(obj1.hrs*3600)+(obj1.min*60)+obj1.sec;int C;
            if(A>B){
                C=A-B;
            }
            else{
                C=B-A;
            }
            time diff(0,0,C);
            return diff;
        }
        void get(){
            cout<<"Enter hours (In Analog Clock Format):"<<endl;
            cin>>hrs;
            if(hrs<0||hrs>12){
                cout<<"Invalid Input, not in analog clock format."<<endl;exit(0);
            }
            cout<<"Enter minutes:"<<endl;
            cin>>min;
            cout<<"Enter seconds:"<<endl;
            cin>>sec;
            if(min<0||min>12){
                cout<<"Invalid Input, not in analog clock format."<<endl;exit(0);
            }
            if(min==12){min=0;}
            if(sec<0||sec>12){
                cout<<"Invalid Input, not in analog clock format."<<endl;exit(0);
            }
            if(sec==12){sec=0;}
        }
        void show(){
            cout<<"Time :  "<<hrs<<":"<<min<<":"<<sec<<endl;
        }
};

int main(){
    time t1;
    t1.get();
    t1.show();
    time t2;
    t2.get();
    t2.show();
    time t3 = t1.difference(t2);
    t3.show();
}
```

Q5:-

```
#include<iostream>
```

```
using namespace std;

int main(){
    int LastDigit = 4;
    int num,remain,digit,samecount=0;
    do{
        cout<<endl<<"Enter Number : "<<endl;
        cin>>num;
        if(num<0 || num>9999){
            cout<<"Invalid Input, Enter a Valid Integer.";
        }
    }while(num<0 || num>9999);
    cout<<endl;
    remain=num;
    while(remain>0){
        digit = remain%10;
        remain = remain/10;
        if(digit == LastDigit){
            samecount++;
        }
    }
    cout<<"In "<<num<<" there are "<<samecount<<" digits that are equal to your
last digit "<<LastDigit<<endl;

}
```

Q6:-

```
#include<iostream> //1 extra iteration that shows a value in 3.14 form
#include<cmath>
#include<iomanip>
using namespace std;

int main(){
    int Term=1;
    double PI_approx=4;
    cout<<"TERM "<<Term<<" || PI = "<<PI_approx<<endl;
    while(PI_approx<3.14 || PI_approx>=3.15){
        PI_approx = PI_approx + (pow(-1,Term)*4)/(2*Term+1);
        Term++;
        cout<<"TERM "<<Term<<" || PI = "<<PI_approx<<endl;
    }
    cout<<"It took "<<Term<<" Terms to reach a value of 3.14 using the
series"<<endl;
}
```

Q7:-

```
#include<iostream>
using namespace std;
int main(){
    int num,reverse=0,remain,digit;
    do{
        cout<<endl<<"Enter Number to check for palindrome : "<<endl;
        cin>>num;
        if(num<0 || num>99999){
            cout<<"Invalid Input, Enter a Valid Integer.";
        }
    }while(num<0 || num>99999);
    remain=num;
    while(remain>0){
        digit = remain%10;
        remain = remain/10;
        reverse = reverse*10+digit;
    }
    cout<<"Number : "<<num<<endl<<"Reverse : "<<reverse<<endl;
    if(num==reverse){
        cout<<"Number is a Palindrome!"<<endl;
    }
    else{
        cout<<"Number is NOT a Palindrome!"<<endl;
    }
}
```

Q8:-

```
#include<iostream>
using namespace std;
//Question 8
int main(){
    int
x=10,y=10,SumOfNumbers,firstNumber=2,secondNumber=3,number=5,largest=2;  //initia
lizations for variables used in question just so this code can work
    //a
    int value;
    cin>>value;
    //b
    cout<<"The product of "<<x<<" and "<<y<<" is "<<x*y<<endl;
    //c
    SumOfNumbers = firstNumber + secondNumber;
    //d
    if(number>=largest){
```

```
        largest = number;
    }
    //e
        /*program to determine he largest of three integers*/
    //f
    int anInteger;
    cin>>anInteger;
    //g
    cout<<"Remainder of "<<x<<" divided by "<<y<<" is "<<x%y<<endl;
    //h
    if(x==y){
        cout<<x<<" is equal to "<<y<<endl;
    }
    //i
    cout<<"The sum is "<<x+y<<endl;
    //j
    cout<<"The value you entered is: "<<value<<endl;
    //k
    int X=1,total=0;
    while(X<=10){
        total+=X;
        ++X;
    }
    //l
    while(X<=100){
        total+=X;
        ++X;
    }
    //m
    while(y>0){
        cout<<y<<endl;
        --y;
    }
    //n
    for(x=100;x>=1;--x){
        cout<<x<<endl;
    }
    //o
    switch(value%2){
        case 0:
            cout<<"Even Integer"<<endl;
            break;
        case 1:
            cout<<"Odd Integer"<<endl;
    }
    //p
    int intVAL;
    char charVAL;
    cin>>intVAL;
```

```
    cin>>charVAL;
    cout<<"Integer : "<<intVAL<<" Character :"<<charVAL<<endl;
    //q
    double p;
    for(p=0.000001;p==0.0001;p+=0.000001){
        cout<<p<<endl;
    }
    //r
    for(x=999;x>=1;x-=2){
        cout<<x;
    }
    //s
    int counter=2;
    do{
        if(counter%2==0){
            cout<<counter<<endl;
        }
        counter+=2;
    }while(counter<100);
    //t
    total=0;
    for(x=100;x<=150;x++){
        total+=x;
    }
}
```