OBJECT ORIENTED PROGRAMMING

LAB # 4

ASSIGNMENT

NAME: S.M.IRTIZA
ROLL NO. : 22K-4638
CLASS: 2-F

QUESTION # 01:

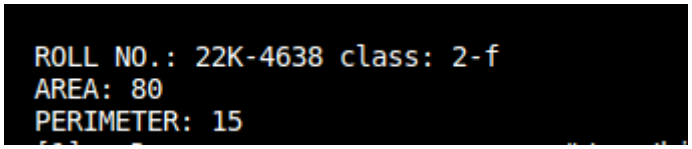
Your task is to create a trapezoid constructor that creates a trapezoid with two sides and height provided by arguments. The trapezoid constructed must have two getters getArea() (area=(a+b)h/2) and getPerimeter() with formula (P=a+b+c+d) which give both respective areas and perimeter.

CODE:

```
////ROLL NO.: 22K-4638 class: 2-f
#include<iostream>
using namespace std;
class Trapezoid{
float side1;
float side2;
float side3;
float side4;
float height;

public:

Trapezoid(){}

Trapezoid(float a,float b, float c){
side1=a;
side2=b;
height=c;
}

float getArea(){
return (height*(side1*side2)/2);
}
void setSide3(float a) {
side3 = a;
}

void setSide4(float b) {
side4 = b;
}

float getSide3() {
return side3;
}

float getSide4() {
return side4;
}
float getPerimeter() {
return (side1+side2+side3+side4);
}
```

```
};

int main() {
cout<<"ROLL NO.: 22K-4638 class: 2-f"<<endl;
Trapezoid t1(5, 4, 8);
t1.setSide3(4);
t1.setSide4(2);
cout << "AREA: " << t1.getArea() << endl;
cout << "PERIMETER: " << t1.getPerimeter() << endl;
return 0;
}
```

OUTPUT:



QUESTION # 02:

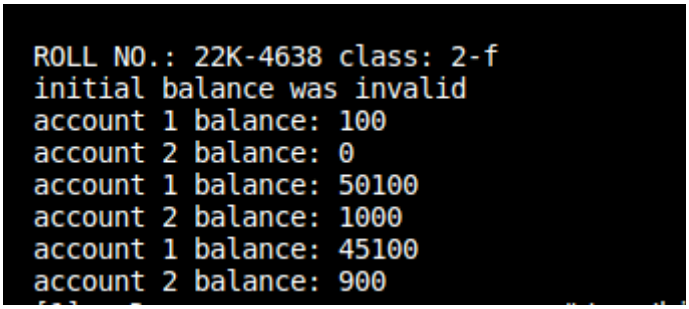
Create an Account class to represent a bank's client bank accounts. Include a data member to reflect the balance of the account. Give each member three functions. The member function credit must be added to the current balance. Member function debit should make a withdrawal from the Account. The get Balance member function should return the current balance.

CODE:

```
////ROLL NO.: 22K-4638 class: 2-f
#include<iostream>
#include<string>
using namespace std;
class BankAccount{
private:
int accountBalance;
public:
BankAccount(){}
BankAccount(int initialBalance){
if(initialBalance>=0){
accountBalance=initialBalance;
}
else{
accountBalance=0;
cout<<"initial balance was invalid"<<endl;
}
}

int getAccountBal(){
return accountBalance;
}

void credit(int amount ){
accountBalance+=amount;
}

void debit(int amount){
if(amount> accountBalance){
cout<<"debit amount exceed the account balance"<<endl;
}
else{
accountBalance-=amount;
}
}
};

int main(){
cout<<"ROLL NO.: 22K-4638 class: 2-f"<<endl;
BankAccount account1(100);
BankAccount account2(-120);
```

```
cout<<"account 1 balance: "<<account1.getAccountBal()<<endl;
cout<<"account 2 balance: "<<account2.getAccountBal()<<endl;

account1.credit(50000);
account2.credit(1000);

cout<<"account 1 balance: "<<account1.getAccountBal()<<endl;
cout<<"account 2 balance: "<<account2.getAccountBal()<<endl;

account1.debit(5000);
account2.debit(100);

cout<<"account 1 balance: "<<account1.getAccountBal()<<endl;
cout<<"account 2 balance: "<<account2.getAccountBal()<<endl;
return 0;
}
```

OUTPUT:



QUESTION # 03:

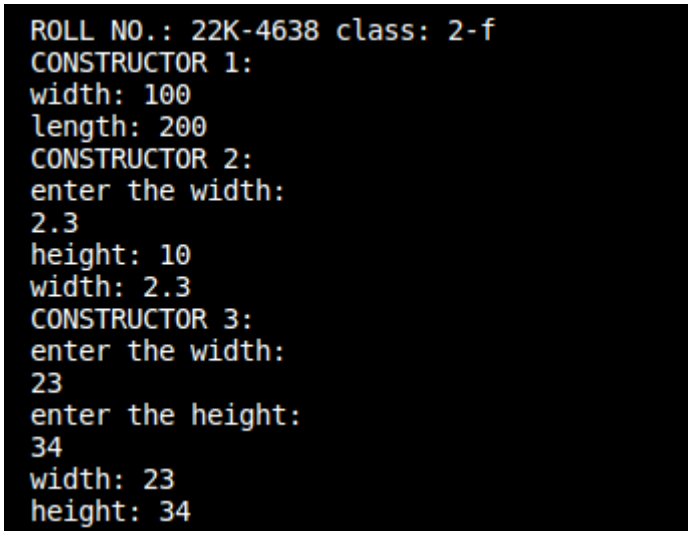
Create a "Box" class given with private members with float values of width and length and implement the constructors as
per the following requirements:
a) A constructor that receives both height and width as parameter to create new Box object only if width and height are both positive values
b) A constructor that receives only height as parameter and takes width as input from the user
c) A constructor that receives no parameter and takes both width and height as user input

CODE:

```
//ROLL NO.: 22K-4638 class: 2-f
#include<iostream>
using namespace std;
class Box{
float width=0;
float length=0;
float height=0;

public:

Box(){
cout<<"enter the width:"<<endl;
cin>>width;

cout<<"enter the height:"<<endl;
cin>>height;

cout<<"width: "<<width<<endl;
cout<<"height: "<<height<<endl;
}

Box(float a,float b){
if(a>=0){
width=a;
cout<<"width: "<<width<<endl;
}
else{
cout<<"enter the positive width: "<<endl;
}

if(b>=0){
length=b;
cout<<"length: "<<length<<endl;
}
else{
cout<<"enter the positive length: "<<endl;
}


}
```

```
Box(float a){
height=a;
cout<<"enter the width:"<<endl;
cin>>width;

cout<<"height: "<<height<<endl;
cout<<"width: "<<width<<endl;
}

};

int main(){
cout<<"ROLL NO.: 22K-4638 class: 2-f"<<endl;
cout<<"CONSTRUCTOR 1: "<<endl;
Box B1(100,200);
cout<<"CONSTRUCTOR 2: "<<endl;
Box B2(10);
cout<<"CONSTRUCTOR 3: "<<endl;
Box B3;
return 0;
}
```

OUTPUT:



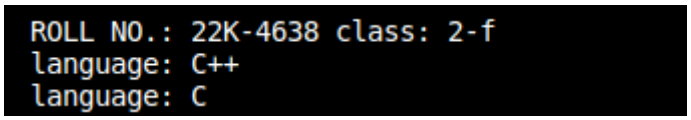
QUESTION # 04

Create a class that contains a String variable named language. 1) Create a default constructor that sets the language to “C++”. 2) Create a parameterized constructor that sets this variable. 3) Create a function that contains a print statement that displays the languages. 4) In the main function create 2 objects. One with the default constructor and one with the parameterized constructor with the value “C”.5) Display the result.

CODE:

```
////ROLL NO.: 22K-4638 class: 2-f
#include<iostream>
#include<string>
using namespace std;
class Class{
string language;

public:
Class(){
language="C++";
}

Class(string a){
language= a;
}
string getLanguage(){
return language;
}
void display(){
cout<<"language: "<<getLanguage()<<endl;
}
};

int main(){
cout<<"ROLL NO.: 22K-4638 class: 2-f"<<endl;
Class C;
C.display();

Class C1("C");
C1.display();
return 0;
}
```

OUTPUT:



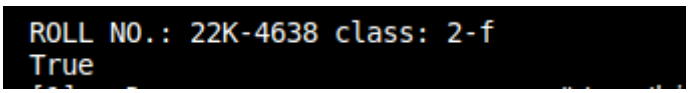
QUESTION #05

Write a class called Holiday. This class has three instance variables: name, Day & month. a) Write a constructor for the class Holiday, which takes a String representing the name, an int representing the day, and a String representing the month as its arguments, and sets the class variables to these values. b) Write a method inSameMonth, which compares two days of the class Holiday, and returns the Boolean value true if they have the same month, and false if they do not. c) Create a Holiday instance with the name “Independence Day”, with the day “14”, and with the month “August”.

CODE:

```
////ROLL NO.: 22K-4638 class: 2-f
#include<iostream>
#include<string>
using namespace std;
class Holiday{
string name;
int Day;
string month;

public:
Holiday(){}
Holiday(string a , int b, string c){
name=a;
Day=b;
month=c;
}

bool inSameMonth(Holiday H1){
if(H1.Day==Day)
{
return true;
}
else{
return false;
}
}
};

int main(){
cout<<"ROLL NO.: 22K-4638 class: 2-f"<<endl;
Holiday H1("Independence Day",14,"August");
Holiday H2("kashmir", 14, "Feburary");
bool x;
x=H2.inSameMonth(H1);
if(x==true)
cout<<"True"<<endl;
else
cout<<"False"<<endl;
return 0;
}
```

OUTPUT:



QUESTION #06

Create A class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables - a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initialize the four instance variables. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiples the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0.

CODE:

```
////ROLL NO.: 22K-4638 class: 2-f
#include<iostream>
#include<string>
using namespace std;
class Invoice{
string partNumber;
string partDescription;
int quantity;
double price;
public:
Invoice(string a,string b, int c, double d){
setPartNumber(a);
setPartDescription(b);
if(c>=0)
setQuantity(c);
else
setQuantity(0);

if(d>=0)
setPrice(d);
else
setPrice(0.0);

}

void setPartNumber(string a){
partNumber=a;
}
void setPartDescription(string b){
partDescription=b;
}
void setQuantity(int c){
if(c>=0)
quantity=c;
else
quantity=0;
}
void setPrice(double d){
if(d>=0)
```

```
price=d;
else
price=0.0;
}
string getPartNumber(){
return partNumber;
}
string getPartDescription(){
return partDescription;
}
int getQuantity(){
return quantity;
}
double getPrice(){
return price;
}
double getInvoiceAmount(){
return (getPrice()*getQuantity());
}
};
int main(){
cout<<"ROLL NO.: 22K-4638 class: 2-f"<<endl;
Invoice in1("k22", "large",10, 120.23 );
cout<<"the price is: "<<in1.getInvoiceAmount()<<endl;
Invoice in2("k22", "large",-10, 120.23 );
cout<<"the price is: "<<in2.getInvoiceAmount()<<endl;
Invoice in3("k22", "large",5, -50 );
cout<<"the price is: "<<in3.getInvoiceAmount()<<endl;
}
```

OUTPUT: