OOP Lab – 05

In lab done: Q1,Q3
After lab done: Q2,Q4,Q5

Name: M. Muzammil Siddiqui

Roll no: 23K-2001

Q1:

```cpp
//23K2001 Muzammil Q1

#include<iostream>

using namespace std;

class realtorCommission{

    private:

        double price,rate,commission;


    public:

        realtorCommission(double price,double rate)

        {

            this->price = price;

            this->rate = rate;

            commission = price * rate;

        }

        realtorCommission(int price,int rate)

        {

            this->price = price;

            this->rate = rate;

            commission = (price * rate)/100;

        }

        void display()

        {

            cout<<"\nPrice: "<<price<<endl;

            cout<<"Rate: "<<rate<<endl;

            cout<<"Commission Earned: "<<commission<<endl;
```

```
        }


};
int main()
{
    realtorCommission shandaar(4500.354,0.75);
    cout<<"\nDisplaying Shaandaar Object: (Decimal values)"<<endl;
    shandaar.display();


    realtorCommission nobel(40000,3);
    cout<<"\nDisplaying Nobel Object: (WholeNumber Values & Commission
through %)"<<endl;
    nobel.display();
    return 0;
}
```

Q2:

```
//23K2001 Muzammil Q2
#include<iostream>
using namespace std;
class carRent{
    private:
    float perday = 50.75;
    const string name;
    int days;
    double rent;

    public:
    carRent(string name,int days) : name(name), days(days)
    {
        rent = days*perday;
    }
    void withDiscount()
    {
        if(days>7){
        cout<<"Congrats you got one-day off discount! Previous rent:
"<<rent<<endl;
        rent-=perday;
        cout<<"After discount, rent: "<<rent<<endl;
        }
        else{
        cout<<"Sorry, no discount applicable!"<<endl;
        withoutDiscount();
```

```
        }
    }
    void withoutDiscount()
    {
        cout<<"Your rent: "<<rent<<endl;
    }
    void display() const{
        cout<<"\nName: "<<name<<endl;
        cout<<"Days: "<<days<<endl;
        cout<<"Rent amount: "<<rent<<endl;
    }
};

int main()
{
    carRent toyota("Mr. Ahmed",5);
    carRent honda("Ms. Noor",18);

    cout<<"Object with <7 days:"<<endl;
    toyota.withDiscount();
    toyota.display();

    cout<<"\nObject with >7 days:"<<endl;
    honda.withDiscount();
    honda.display();
    return 0;
}
```

Q3:

```
//23K2001 Muzammil Q3
#include<iostream>
using namespace std;
class circle{
    private:
        float radius;
        const double PI= 3.141529654;

    public:
        circle()
        {
            radius = 0;
        }
        float calcCircumference()
        {
            return 2*PI*radius;
        }
        float calcArea()
        {
            return PI*radius*radius;
        }
        void setR(float radius)
        {
            this->radius = radius;
```

```
        }
        float getR()
        {
            return radius;
        }
        void show()
        {
            cout<<"\nRadius : "<<radius<<" units."<<endl;
            cout<<"Area : "<<calcArea()<<" sq-units."<<endl;
            cout<<"Circumference: "<<calcCircumference()<<"
units."<<endl;
        }

};
int main()
{
    circle one;
    cout<<"Circle one (by default constructor): "<<endl;
    one.show();
    circle two;
    two.setR(7.5);
    cout<<"\nCircle two (by parameter constructor): "<<endl;
    two.show();
    return 0;
}
```

Q4:

```
//23K2001 Muzammil Q4
#include<iostream>
using namespace std;
class book{
private:
string title;
float price;
public:
book():title(""),price(0.0){}
book(string title, float price):title(title),price(price){}
void setTitle(string title)
{
    this->title = title;
}
void setPrice(float price)
{
    this->price = price;
}
string getTitle()
{
    return title;
}
float getPrice()
{
    return price;
```

```
}
};
class payment{
private:
float amount;
public:
void setAmount(float a)
{
    amount = a;
}
float getAmount()
{
    return amount;
}
};


class inventory{
private:
book *books;
int n=0;


public:
inventory() {
books = nullptr;
}
book* getBooks() const{
    return books;
```

```
}
int getNumBooks() const
{
    return n;
}
void addBook(const book& newbook)
{
    int i;
    n++;
    book *b = new book[n];
    for(i=0;i<n-1;i++)
    {
    b[i] = books[i];
    }
    delete[] books;
    books = b;
    books[n-1] = newbook;
    cout<<"\nBook added successfully!"<<endl;
}


void removeBook()
{
    int x,i;
    if(displayInv()==true){
        cout<<"Please select which book# to remove? ";
        cin>>x;
        if(x > 0 && x<=n)
```

```
        {
            for(i=x-1;i<n-1;i++)
            {
                books[i] = books[i+1];
            }

            n--;
            book* newBooks = new book[n];
            for(int i = 0; i < n; i++)
            {
                newBooks[i] = books[i];
            }
            delete[] books;
            books = newBooks;
            cout << "\nBook removed successfully!" << endl;
        }
        else
        cout<<"\nInvalid # , please try again!"<<endl;
    }
}
bool displayInv()
{
    if(n!=0)
    {
        int i;
        cout<<"\n#    Title:\tPrice:"<<endl;
        for(i=0;i<n;i++)
```

```
        {
            cout<<i+1;
            cout<<"
"<<books[i].getTitle()<<"\t"<<books[i].getPrice()<<endl;
        }
        return true;
    }
    else{
        cout<<"\nNo books to display in inventory, please add some
first!"<<endl;
        return false;
    }
}

~inventory()
{
    delete[] books;
}
};

class order{
private:
int x;
book *b;
payment p;

public:
order(int n) : x(n){
```

```
    b = new book[x];
}
void placeOrder(const inventory& inv)
{
    int i,j;
    int booknum[x];
    int booksinInv = inv.getNumBooks();
    float amount=0;
    cout<<"\nPlease select which book# to order? ";
    for(i=0;i<x;i++)
    {
        cout<<"\nBook "<<i+1<<": ";
        cin>>booknum[i];
        if(booknum[i]>booksinInv || booknum[i]==0)
        {
            cout<<"\nInvalid # , there is no
book#"<<booknum[i]<<"!!!"<<endl;
            cout<<"Kindly select again: ";
            i--;
            continue;
        }
        cout<<"\nSelected Book#"<<booknum[i]<<endl;
        booknum[i]-=1;
    }
    for(i=0;i<x;i++)
    {
        b[i]=(inv.getBooks())[booknum[i]];
```

```
        amount+=b[i].getPrice();
    }
    p.setAmount(amount);
    displayOrderinfo();
    cout<<"\nTotal payment: "<<p.getAmount()<<endl;
    cout<<"Order placed successfully!"<<endl;
}
void displayOrderinfo()
{
    int i;
    cout<<"\nUser ordered "<<x<<" book/books:"<<endl;
    cout<<"\n#    Title:\tPrice:"<<endl;
    for(i=0;i<x;i++)
    {
        cout<<i+1<<"    "<<b[i].getTitle()<<"\t"<<b[i].getPrice()<<endl;
    }
}
~order(){
    delete[] b;
}
};

int main()
{
    inventory fastInventory;
    cout<<"\t\tWelcome to FAST INVENTORY SYSTEM\n"<<endl;
    int ch;
```

```
    do {
        cout<<"\nMenu:"<<endl;
        cout<<"1. Add Book"<<endl;
        cout<<"2. Remove Book"<<endl;
        cout<<"3. Display Entire Inventory"<<endl;
        cout<<"4. Place Order"<<endl;
        cout<<"5. Exit"<<endl;
        cout<<"Input choice: ";
        cin>>ch;
        switch(ch){
            case 1:
            {
                string title;
                float p;
                cout<<"Input title: ";
                cin>>title;
                cout<<"Input price: ";
                cin>>p;
                book b(title,p);
                fastInventory.addBook(b);
                break;
            }
            case 2:
            {
                fastInventory.removeBook();
                break;
            }
```

```
            case 3:
            {
                fastInventory.displayInv();
                break;
            }
            case 4:{
                if(fastInventory.displayInv()==true)
                {
                    int n;
                    cout<<"How many books you want to order: ";
                    cin>>n;
                    order recent(n);
                    recent.placeOrder(fastInventory);
                    break;
                }
                else
                break;
            }
            case 5:
                return 0;
            default:
                cout<<"Invalid input!"<<endl;
        }
    } while(ch!=5);
    return 0;

}
```

Outputs for Question#4:

```
                Welcome to FAST INVENTORY SYSTEM


Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 3

No books to display in inventory, please add some first!

Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 2

No books to display in inventory, please add some first!
```

```
Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 1
Input title: Calculus
Input price: 900

Book added successfully!

Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 1
Input title: LearningC++
Input price: 400

Book added successfully!
```

```
Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 2

#    Title:     Price:
1    Calculus   900
2    Urdu       200
3    Magazine   600
4    English    200
5    Physics    450
Please select which book# to remove? 3

Book removed successfully!
```

```
Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 3

#    Title:     Price:
1    Calculus   900
2    Urdu       200
3    English    200
4    Physics    450
```

```
#    Title:     Price:
1    Calculus   900
2    Urdu       200
3    English    200
4    Physics    450
Please select which book# to remove? 5

Invalid # , please try again!

Menu:
1. Add Book
2. Remove Book
3. Display Entire Inventory
4. Place Order
5. Exit
Input choice: 2

#    Title:     Price:
1    Calculus   900
2    Urdu       200
3    English    200
4    Physics    450
Please select which book# to remove? 0

Invalid # , please try again!
```

```
Input choice: 4

#    Title:     Price:
1    Calculus   900
2    Urdu       200
3    English    200
4    Physics    450
How many books you want to order: 2

Please select which book# to order?
Book 1: 5

Invalid # , there is no book#5!!!
Kindly select again:
Book 1: 4

Selected Book#4

Book 2: 0

Invalid # , there is no book#0!!!
Kindly select again:
Book 2: 1

Selected Book#1

User ordered 2 book/books:

#    Title:     Price:
1    Physics    450
2    Calculus   900

Total payment: 1350
Order placed successfully!
```

Q5:

```
//23K2001 Muzammil Q5
#include<iostream>
using namespace std;
class book{
private:
string title, author,isbn,genre;
bool status;
public:
book(string t,string a,string i,string g,bool s=true)
{
    title = t;
    author = a;
    isbn = i;
    genre = g;
    status = s;
}
book(){}
void displayBookinfo()
{
    cout<<"Title: "<<title<<endl;
    cout<<"Author: "<<author<<endl;
    cout<<"ISBN: "<<isbn<<endl;
```

```
    cout<<"Genre: "<<genre<<endl;
    cout<<"Availablity: ";
    if(status==true)
    cout<<"Available"<<endl;
    else
    cout<<"Issued"<<endl;
}
string getTitle(){
    return title;
}
string getAuthor(){
    return author;
}
string getisbn(){
    return isbn;
}
string getGenre(){
    return genre;
}
void setStatus(bool s)
{
    status = s;
}
bool getStatus() const {
```

```
    return status;
}
};

void displayBooks(book b[],int n)
{
    int i;
    cout<<"# "<<"Title:\t\t"<<"\tStatus:"<<endl;
    for(i=0;i<n;i++)
    {
        cout<<i+1<<"
"<<b[i].getTitle()<<"--------------------(";
        if(b[i].getStatus()==true)
        cout<<"Available)"<<endl;
        else
        cout<<"Issued)"<<endl;
    }
}

class loan{
private:
int due;

public:
```

```
loan(int d=20240101){due = d;}


void setDuedate(int d)
{
    due = d;
}
int getDuedate()
{
    return due;
}
};


class fine{
    private:
    float finemoney=0;
    public:
    void checkforfine(loan l,int d)
    {
        if(d>l.getDuedate())
        {
        finemoney+=500;
        cout<<"Patron has to pay fine of Rs:"<<finemoney<<"
for late return!"<<endl;
        }
```

```
    }
};


class patron{
private:
string name;
int n;
book** borrowed;
loan* loans;
fine f;


public:
patron(){borrowed = nullptr;
loans = nullptr;
n=0;}
patron(string name):name(name){
    borrowed = nullptr;
    loans = nullptr;
    n=0;
}
~patron(){delete[] borrowed;
delete[] loans;}


void borrowBook(book& b)
```

```
{
    int i;


    if ((borrowed == nullptr && loans == nullptr) &&
b.getStatus()==true) {
        n++;
        borrowed = new book*[n];
        loans = new loan[n];
        cout<<"\nSet due date for returning this book
(YYYY/MM/DD): ";
        int d;
        cin>>d;
        loans[0].setDuedate(d);
        b.setStatus(false);
        borrowed[0] = &b;
        cout<<"\nBook borrowed successfully by
"<<name<<"!"<<endl;
    }


    else if(b.getStatus()==true)
    {
        n++;
        book** updated = new book*[n];
        loan* l = new loan[n];
        for(i=0;i<n-1;i++)
```

```
        {
            updated[i] = borrowed[i];
            l[i]=loans[i];
        }
        delete[] borrowed;
        delete[] loans;
        cout<<"\nSet due date for returning this book
(YY/MM/DD): ";
        int d;
        cin>>d;
        b.setStatus(false);
        updated[n-1] = &b;
        borrowed = updated;
        loans = l;
        loans[n-1].setDuedate(d);
        cout<<"\nBook borrowed successfully by
"<<name<<"!"<<endl;
    }
    else
    cout<<"This book is not available, please select some
other."<<endl;
}
void returnBook()
{
    cout<<endl;
```

```
    if(displayBorrowed()==true)
    {
    int x,i;
    cout<<"\nSelect which book to return: ";
    cin>>x;
    if(x>0 && x<=n)
    {
        int today;
        cout<<"Input today's date (YYYY/MM/DD): ";
        cin>>today;
        f.checkforfine(loans[x-1],today);
        borrowed[x-1]->setStatus(true);
        for(i=x-1;i<n-1;i++)
        {
            borrowed[i]=borrowed[i+1];
            loans[i]=loans[i+1];
        }

        n--;
        book** updatedbooks = new book*[n];
        loan* updatedloans = new loan[n];
        for(i=0;i<n;i++)
        {
            updatedbooks[i] = borrowed[i];
```

```
            updatedloans[i] = loans[i];
        }

        delete[] borrowed;
        delete[] loans;
        borrowed = updatedbooks;
        loans = updatedloans;
        cout<<"\nBook returned successfully by
"<<name<<"!"<<endl;
    }
    else
        cout<<"\nInvalid # , please try again!"<<endl;
    }
}
void displayPatroninfo()
{
    cout<<"Name: "<<name<<endl;
    cout<<"Borrowed books: "<<endl;
    displayBorrowed();
}
bool displayBorrowed()
{
    if(n>0)
    {
```

```
        int i;
        for(i=0;i<n;i++)
        {
            cout<<i+1<<"
"<<borrowed[i]->getTitle()<<"-----\tDue Date (YY/MM/DD):
"<<loans[i].getDuedate()<<endl;
        }
        return true;
    }
    else
    {
        cout<<"\nNo books to display for Patron: "<<name<<",
please issue some first!"<<endl;
        return false;
    }
}
string getName()
{
    return name;
}
};

void trackLoans(patron **p, int patrons)
{
    int i;
```

```
    for(i=0;i<patrons;i++)
    {
        cout<<"\nPatron#"<<i+1<<":"<<endl;
        p[i]->displayPatroninfo();
    }
}

int main()
{
    patron m("Muzammmil");
    patron q("Qasim");
    patron *patrons[]={&m,&q};
    book bookshelf[6]={book("Mathematics","Sindh
Board","MT2001029FS","Academic"),book("Urdu","Faiz","SS1008297FS","Research",true),

book("Calculus","Anton","MT8400229SS","Academic",false),book("Fundamentals of
Physics","Pearlson","EE1002534SS","Academic",true),book("LearningC++","Oxford","CS22131029SS","Academic"),book("DeepWater","Cambridge","SS4301029GN","Literature")};

    int ch;
    do {
        cout<<"\nMenu:"<<endl;
        cout<<"1. Borrow a book"<<endl;
```

```
        cout<<"2. Return a book"<<endl;
        cout<<"3. Display loans"<<endl;
        cout<<"4. Display books"<<endl;
        cout<<"5. Exit"<<endl;
        cout<<"Input choice: ";
        cin>>ch;

        switch (ch) {
            case 1:
            {
                int u;
                cout<<"Patrons:"<<endl;
                cout<<"1. "<<m.getName()<<"\n2.
"<<q.getName()<<endl;
                cout<<"\nPlease select for which patron: ";
                cin>>u;

                if(u==1)
                {
                    cout<<"Which book "<<m.getName()<<" wants
to borrow?"<<endl;
                    displayBooks(bookshelf,6);
                    int s;
                    cin>>s;
```

```
                    if(s>6 || s<1)
                        cout<<"\nInvalid # , please try
again!"<<endl;
                    else
                    m.borrowBook(bookshelf[s-1]);
                }
                else if(u==2)
                {
                    cout<<"Which book "<<q.getName()<<" wants
to borrow?"<<endl;
                    displayBooks(bookshelf,6);
                    int s;
                    cin>>s;
                    if(s>6 || s<1)
                        cout<<"\nInvalid # , please try
again!"<<endl;
                    else
                        q.borrowBook(bookshelf[s-1]);
                }
                else
                cout<<"Invalid input."<<endl;

                break;

            }
```

```
            case 2:
            {
                int u;
                cout<<"Patrons:"<<endl;
                cout<<"1. "<<m.getName()<<"\n2.
"<<q.getName()<<endl;
                cout<<"\nPlease select for which patron: ";
                cin>>u;

                if(u==1)
                {
                    m.returnBook();
                }
                else if(u==2)
                {
                    q.returnBook();
                }
                else
                cout<<"Invalid input."<<endl;
                break;
            }
            case 3:
            {
                trackLoans(patrons,2);
```

```
                break;
            }
            case 4: {
                displayBooks(bookshelf,6);
                break;
            }
            case 5:
                return 0;
            default:
                cout<<"Invalid input!"<<endl;
        }
    } while(ch!=5);
    return 0;
}
```

Outputs for Question#5:

```
Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 4
# Title:                          Status:
1 Mathematics--------------------(Available)
2 Urdu---------------------------(Available)
3 Calculus-----------------------(Issued)
4 Fundamentals of Physics--------------------(Available)

Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 1
Patrons:
1. Muzammmil
2. Qasim

Please select for which patron: 1
Which book Muzammmil wants to borrow?
# Title:                          Status:
1 Mathematics--------------------(Available)
2 Urdu---------------------------(Available)
3 Calculus-----------------------(Issued)
4 Fundamentals of Physics--------------------(Available)
3
This book is not available, please select some other.
```

```
Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 1
Patrons:
1. Muzammmil
2. Qasim

Please select for which patron: 1
Which book Muzammmil wants to borrow?
# Title:                          Status:
1 Mathematics--------------------(Available)
2 Urdu---------------------------(Available)
3 Calculus-----------------------(Issued)
4 Fundamentals of Physics--------------------(Available)
1

Set due date for returning this book (YYYY/MM/DD): 20240310

Book borrowed successfully by Muzammmil!
```

```
Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 3

Patron#1:
Name: Muzammmil
Borrowed books:

No books to display for Patron: Muzammmil, please issue some first!

Patron#2:
Name: Qasim
Borrowed books:

No books to display for Patron: Qasim, please issue some first!

Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 5
(base) PS C:\Users\Lenovo\Desktop\OOP (LAB)\Lab Tasks\Lab 05>
```

```
Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 2
Patrons:
1. Muzammmil
2. Qasim

Please select for which patron: 2

1 Fundamentals of Physics-----  Due Date (YY/MM/DD): 20201003

Select which book to return: 1
Input today's date (YYYY/MM/DD): 20250607
Patron has to pay fine of Rs:500 for late return!

Book returned successfully by Qasim!
```

```
Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 4
# Title:                          Status:
1 Mathematics-------------------(Issued)
2 Urdu--------------------(Issued)
3 Calculus--------------------(Issued)
4 Fundamentals of Physics-------------------(Issued)

Menu:
1. Borrow a book
2. Return a book
3. Display loans
4. Display books
5. Exit
Input choice: 3

Patron#1:
Name: Muzammmil
Borrowed books:
1 Mathematics-----      Due Date (YY/MM/DD): 20240310
2 Urdu-----     Due Date (YY/MM/DD): 20240310

Patron#2:
Name: Qasim
Borrowed books:
1 Fundamentals of Physics-----  Due Date (YY/MM/DD): 20201003
```