

## OOP-LAB Task 4

Roll No: 24P-0706

Dept: BS-CS

Name: Aazan Noor Khuwaja

Section : 2D

**Qns1:** Write a program to define two constructors to find out the maximum values.

Code:

```
#include<iostream>

using namespace std;

class max_value{

private:
    float m_value;

public:
    max_value(float a, float b){
        if(a>b){
            m_value=a;
        }
        else{
            m_value=b;
        }
    }

    max_value(float a, float b, float c){
        if(a>b && a>c){
            m_value=a;
        }
        else if(b>a && b>c){
            m_value=b;
        }
    }
}
```

```

    }
else{
    m_value=c;
}
}

void display(){

    cout<<"Max value: "<<m_value<<endl;
}

};

int main(){

    max_value m1(2.5, 3.7);

    m1.display();

    max_value m2(10.2, 4.3, 5.6);

    m2.display();

    return 0;
}

```

**Output:**

```

PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4> g++ -g qns1.cpp -o ./output
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4> ./output
Max value: 3.7
Max value: 10.2
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4>

```

## Qns 2:

Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50 cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected Model this tollbooth with a class called tollbooth. The two data items are a type unsigned int to hold the total

number of cars, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0. A member function called payingCar() increments the car total and adds 0.50 to the cash total. Another function, called nopayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals.

Make appropriate member functions const.

Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the Esc key should cause the program to print out the total cars and total cash and then exit.

## Code:

```
#include <iostream>
#include <conio.h>
using namespace std;
class toll_booth {
private:
    unsigned int total_cars;
    double total_money;

public:
    toll_booth(){
        total_cars=0;
        total_money=0;
    }
    void paying_car() {
        total_cars++;
        total_money += 0.5;
    }
}
```

```

cout<<"paying car: "<<total_cars<<"\nTotal Money :"<<total_money<<endl;
}

void nopay_car() {
    total_cars++;
    cout<<"non paying car: "<<total_cars<<endl;
}

void display() const {
    cout<<"Final Result"<<endl;
    cout << "Total cars: "<< total_cars << endl;
    cout << "Total money: "<< total_money << endl;
}

unsigned int get_total_cars() const {
    return total_cars;
}

double get_total_money() const {
    return total_money;
}

int main() {
    toll_booth booth;
    char ch;
    cout << "Press 'p' for paying car, 'n' for nonpaying car, and 'Esc' to exit and display totals." << endl;
    while (true) {
        ch=_getch();
        if (ch == 27) {
            booth.display();
            break;
        } else if (ch == 'p' || ch == 'P') {

```

```

    booth.paying_car();

} else if (ch == 'n' || ch == 'N') {

    booth.nopay_car();

}

else{

    cout<<"Input is not correct!"<<endl;

}

}

return 0;

}

```

**Output:**

```

PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4> g++ -g qns2.cpp -o ./output
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4> ./output
Press 'p' for paying car, 'n' for nonpaying car, and 'Esc' to exit and display totals.
paying car: 1
Total Money :0.5
paying car: 2
Total Money :1
paying car: 3
Total Money :1.5
non paying car: 4
non paying car: 5
non paying car: 6
non paying car: 7
paying car: 8
Total Money :2
Final Result
Total cars: 8
Total money: 2
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4>

```

**Qns3:** Create a C++ program to simulate a simple bank account management

system using OOP concepts.

Code:

```

#include<iostream>
using namespace std;

```

```
class bank_acc{  
private:  
    string account_number;  
    string account_holder_name;  
    float balance;  
public:  
    bank_acc(){  
        account_number="N/A";  
        account_holder_name="Unknown";  
        balance=0;  
    }  
    bank_acc(string acc_no, string acc_holder_name, float bal){  
        account_number=acc_no;  
        account_holder_name=acc_holder_name;  
        balance=bal;  
    }  
    ~bank_acc(){  
        cout<<"Account Number:"<<account_number<<" Closed"<<endl;  
    }  
    void deposit(double amount){  
        balance+=amount;  
        cout<<"Amount Deposited: "<<amount<<endl;  
    }  
    void withdraw(double amount){  
        if(balance>amount){  
            balance-=amount;  
            cout<<"Amount Withdrawn: "<<amount<<endl;  
        }  
        else{  
    }
```

```
    cout<<"Not sufficient Balance because we have to leave balance greater than zero"<<endl;
}

void display(){

    cout<<"Account Number: "<<account_number<<endl;
    cout<<"Account Holder Name: "<<account_holder_name<<endl;
    cout<<"Balance: "<<balance<<endl;
}

};

int main(){

    bank_acc b1;

    b1.display();

    bank_acc b2("4520284", "Qasim", 1500);

    b2.display();

    b2.deposit(500);

    b2.withdraw(1800);

    b2.display();

    return 0;
}
```

**Output:**

```
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4> g++ -g qns3.cpp -o ./output
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4> ./output
Account Number: N/A
Account Holder Name: Unknown
Balance: 0
Account Number: 4520284
Account Holder Name: Qasim
Balance: 1500
Amount Deposited: 500
Amount Withdrawn: 1800
Account Number: 4520284
Account Holder Name: Qasim
Balance: 200
Account Number:4520284 Closed
Account Number:N/A Closed
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-4>
```