Muhammad Mufeez

23k-0800

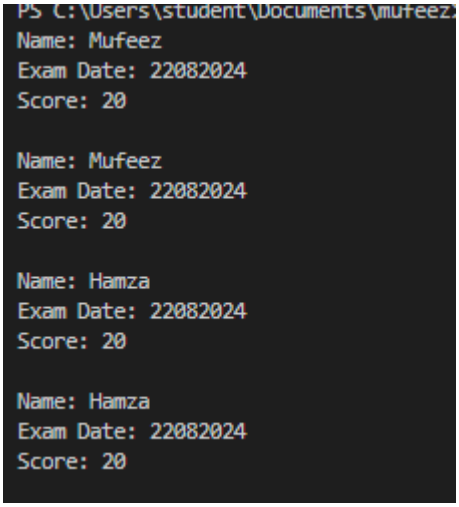
Question 1:

```cpp
#include <iostream>

using namespace std;

class BankAccount
{
    float balance;
public:
    BankAccount()
    {
        cout << balance<<endl;
    }
    BankAccount(float balance)
    {
        this->balance = balance;
        cout << balance <<endl;
    }

    BankAccount(const BankAccount &obj)
    {
        this->balance = obj.balance;
        balance -= 200;
        cout << "Account 3:" << balance <<endl;
        cout << "Account 2:" << obj.balance <<endl;
    }
};
int main()
{
    BankAccount account1;
    BankAccount account2(1000);
    BankAccount account3 = account2;
}
```

```
PS C:\Users\student\Documents\mufeez> cd "c:\Users\
5.88684e-039
1000
Account 3:800
Account 2:1000
```

Question 2:

```cpp
#include <iostream>

using namespace std;

class Exam
{
public:
    string name;
    string date;
    int score;

public:
    void setName(string name)
    {
        this->name = name;
    }
    void setDate(string date)
    {
        this->date = date;
    }
    void setScore(int score)
    {
        this->score = score;
    }
    void displayDetails()
    {
        cout << "Name: " << name << endl;
        cout << "Exam Date: " << date << endl;
        cout << "Score: " << score << endl
             << endl;
    }
};
int main()
{
    Exam *ptr1 = new Exam();
    ptr1->setDate("22082024");
    ptr1->setName("Mufeez");
    ptr1->setScore(20);
    ptr1->displayDetails();

    Exam *ptr2 = ptr1;
    ptr2->displayDetails();

    ptr2->setName("Hamza");
    // THis changes value of name in both of the objects
    ptr2->displayDetails();
    ptr1->displayDetails();

    return 0;
}

//This program does shallow copy, to resolve this issue we have to do deep copy in objects
```


PS C:\Users\student\Documents\mufeez>
Name: Mufeez
Exam Date: 22082024
Score: 20
Name: Mufeez
Exam Date: 22082024
Score: 20
Name: Hamza
Exam Date: 22082024
Score: 20
Name: Hamza
Exam Date: 22082024
Score: 20

Question 3:

```cpp
#include <iostream>

using namespace std;

class Document{
    public:
    string *content;

    Document(const string s){
        content =  new string(s);
    }
    Document(Document &other){
        this->content = new string(*other.content);
    }

    Document& operator=(Document &other){
        this->content = new string(*other.content);
        return *this;
    }

    ~Document(){
        delete content;
        content = nullptr;
    }
};
int main()
{
    Document doc1("This is doc 1");
    Document doc2 = doc1;

    *doc2.content = "This is changed text";
    cout<<*doc1.content<<endl;
    cout<<*doc2.content<<endl;

    return 0;
}
```

```
PS C:\Users\student\Documents\
This is doc 1
This is changed text
PS C:\Users\student\Documents\
```