QUESTION 1:

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

class BankAccount
{
private:
    float balance;

public:
    BankAccount()
    {
        balance = 0.0;
        cout << "Balance: " << balance << endl;
    }
    BankAccount( float bal )
    {
        balance = bal;
        cout << "Balance: " << balance << endl;
    }
    BankAccount( BankAccount& acc )
    {
        balance = acc.balance - 200;
        cout << "Balance: " << balance << endl;
        cout << "Balance: " << acc.balance << endl;
    }
    void PrintBalance()
    {
        cout << "Balance: " << balance << endl;
    }
};

int main()
{
    BankAccount account1;
    BankAccount account2( 1000 );
    BankAccount account3 = account2;
}
```

QUESTION 2:

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

class Exam
{
    string studentName;
    int examDate;
    float score;

public:
    void PrintDetails()
    {
        cout << "Student name: " << studentName << endl;
        cout << "Exam date: " << examDate << endl;
        cout << "Score: " << score << endl;
    }

    void SetStudentName( string name )
    {
        studentName = name;
    }
    void SetExamDate( int date )
    {
        examDate = date;
    }
    void SetScore( float score )
    {
        this->score = score;
    }
};

int main()
{
    /*In current example, generating a shallow copy won't result in potential
issues as we are dealing with basic data types(string, int, float). But if we
deal with pointers or dynamic allocation, shallow copy would result in copying
the same memory address due to which modifying one would change other too.*/

    Exam e1;
    e1.SetStudentName( "Talha" );
    e1.SetExamDate( 12 );
    e1.SetScore( 50.0 );
```

```
    e1.PrintDetails();
    Exam e2 = e1;
    e2.PrintDetails();

    cout << "--------------------" << endl;

    e2.SetExamDate( 13 );
    e2.SetScore( 90.0 );
    e2.PrintDetails();
    e1.PrintDetails();
}
```

QUESTION 3:

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

class Document
{
    const char* textContent;

public:
    //Default Constructor
    Document()
    {
        textContent = " ";
    }

    //Parameterized Constructor
    Document( const char* text )
    {
        textContent = new char[strlen( text ) + 1];
        textContent = text;
    }

    //Destructor
    ~Document()
```

```
    {
        delete textContent;
    }

    //Copy Constructor
    Document( Document& doc )
    {
        textContent = new char[strlen( doc.textContent ) + 1];
        textContent = doc.textContent;
    }

    //Overloading Assignment Operator
    Document& operator=( Document& doc )
    {
        if (this != &doc)
        {
            textContent = new char[strlen( doc.textContent ) + 1];
            textContent = doc.textContent;
        }
    }

    void SetContent( const char* text )
    {
        textContent = text;
    }

    void PrintContent()
    {
        cout << textContent << endl;
    }
};

int main()
{
    const char* text = "Hey, my name is rabia";
    Document doc( text );
    Document doc1 = doc;

    //modifying the original one
    doc.SetContent( "Hey, my name is talha" );

    Document doc3;
    doc3 = doc;
    doc.PrintContent();
    doc1.PrintContent();
```

```
    doc.PrintContent();
    doc3.PrintContent();
}
```