

CSI 2172
Assignment 3
Date: 27 June 2004
Due date: 19 July 2004

Question 1

Define a $n \times n$ matrix class and implement the following operations on this class.

Addition + (Add two matrices. i.e. $M+N=P$ where M, N, P are $n \times n$ matrices)

Substraction - (Subtract two matrices. $M-N=P$ where M,N,P $n \times n$ matrices)

Multiplication * (Multiply a matrix by a scalar. The matrix can be on either side of the * operator; i.e $M*s=s*M=N$ where M, N are matrices and s is a scalar)

Output operator << (to be able to output a matrix with an ostream object such as cout. for example if M and N are $n \times n$ matrices the following operation must be allowed: `cout << M << N`, etc..

Question 2

Write a **recursive** function that lists all of the two-letter subsets for a given set of letters. For example:
[A,'C','E'] => [A,'C'],[A,'E'],[C,'E']

Question 3

Create a base called **Vehicle** that has the manufacturer's name (type string), number of cylinder's in the engine (type int), and owner (type **Person** given below). Then create a class called **Truck** that is derived from Vehicle and has additional properties, the load capacity in tons (type **double** since it may contain a fractional part) and towing capacity in pounds (type int). Be sure your classes have a reasonable complement of constructors and accessor methods, an overloaded assignment operator, and a copy constructor,

Write a driver program that tests all your methods.

The definition of the class **Person** is below. The implementation of the class is part of this programming project.

```
Class Person
{
public:
    Person();
    Person(string theName);
```

```

    Person(const Person& theObject);
    string getName() const;
    Person& operator=(const Person& rtSide);
    friend istream& operator >>(istream& inStream, Person& personObject);
    friend ostream& operator <<(ostream& outStream, const Person& personObject);

```

```

private:
    string name;
};

```

Question 4

Give the definition of two classes, **Patient** and **Billing**, whose objects are records for a clinic. **Patient** will be derived from the class **Person** given below. A **Patient** record has the patient's name (inherited from the class **Person**) and primary physician, of type **Doctor** defined in Programming Project 3 (please see below). A **Billing** object will contain a **Patient** object and a **Doctor** object, and an amount due of type double. Be sure your classes have a reasonable complement of constructors and accessor methods, an overloaded assignment operator, and a copy constructor. First write a driver program to test all your methods, then write a test program that creates at least two patients, at least two doctors, at least two **Billing** records, then prints out the total income from the **Billing** records.

```

class Person
{
public:
    Person();
    Person(string theName);
    Person(const Person& theObject);
    string getName() const;
    Person& operator=(const Person& rtSide);
    friend istream& operator >>(istream& inStream, Person& personObject);
    friend ostream& operator <<(ostream& outStream, const Person& personObject);

```

```

private:
    string name;
};

```

** project 3: give the definition of a class named **Doctor** whose objects are records for a clinic's doctors. This class will be a derived class of the class **SalariedEmployee** given below. A **Doctor** record has the doctor's specialty (such as "Pediatrician," "Obstetrician," "General Practitioner," etc., so use type string), and office visit fee (use type double). Be sure your class has a reasonable complement of constructors and accessor methods, an overloaded assignment operator, and a copy constructor. *Write a driver program to test all your methods.*

```

class Employee
{

```

```

public:
    Employee();
    Employee(string theName, string theSsn);
    string getName() const;
    string getSsn() const;
    double getNetPay() const;
    void setName(string newName);
    void setSsn(string newSsn);
    void setNetPay(double newNetPay);
    void printCheck() const;
private:
    string name;
    string ssn;
    double netPay;
}

class SalariedEmployee : public Employee
{
public:
    SalariedEmployee();
    SalariedEmployee (string the Name, string theSsn, double theWeeklySalary);
    double getSalary() const;
    void setSalary(double newSalary);
    void printCheck();
private:
    double salary; //weekly
}

```

For Q4, first, please implement a class named Doctor. Please mention that you need not write a drive program to test all your methods in project3 (in red). Then do question 4.

Good luck!