Programmieren in C++ Sommersemester 1997
Klausur, 3.7.1997

Vorname	Name	Matr.-Nr.	Email	Bonus	Klausur	Summe
				/7.5	/50	

1. Was ist ... (Bitte knappe, präzise Antworten) [10 Punkte]

a) Eine abstrakte Klasse?

b) Eine Schablonenfunktion?

c) Eine virtuelle Basisklasse?

d) Ein Klassenattribut?

e) Dynamische Bindung?

2. Welche Ausgabe erzeugt ... [10 Punkte]

```
#include <iostream.h>

class bruch
{
private:
    int nenn;
    int zaehl;
public:
    bruch(int z=0, int n=1): zaehl(z), nenn(n) {test('K');}
    bruch(const bruch& b): nenn(b.nenn), zaehl(b.zaehl) {test('C');}
    ~bruch() {test('D');}
    void test(char w) {cout <<  w << ": " << zaehl << '/' << nenn << endl;}
};

bruch gen7()
{
    bruch result(7,1);
    return result;
}

int main()
{
    bruch cb(1);
    bruch ba[3] = {bruch(2,3), cb};
    bruch *bp = new bruch[2];
    gen7().test('M');
}
```

3. Ergänzen Sie ... [10 Punkte]

```
class vector
{   double *co;
    int dim;
public: vector(int d) : dim(d), co(new double[d]) {}
        double& operator [] (int d) {return co[d];}
};

int main()
{ vector v(10);
  for (int i = 1; i <= 10; i++)
    v[i-1] = i;
  double sum = v;
  cout << sum << endl;
}
```

An der Stelle `double sum = v;` meldet der Compiler:

```
Error: Cannot use vector to initialize double.
```

Ergänzen Sie die Klasse `vector`, so daß dieser Fehler nicht mehr auftritt und `sum` mit der Summe der Elemente im Vektor `v` (hier 55) initialisiert wird.

4. Implementieren Sie ... [10 Punkte]

die Klasse `Termin`, die wie folgt entworfen worden ist:

- Es existiert eine Klasse `Uhrzeit`

```
class Uhrzeit
{ private: ...
  public: Uhrzeit();
          Uhrzeit (int std, int min);
          void print();
};
```

- Es existiert eine Klasse `Date`

```
class Date
{ private: ...
  public: Date();
          Date (int tag, int monat, int jahr);
          void print();
};
```

- Ein `Termin` *ist ein* `Date`.
- Ein `Termin` *hat eine* `Uhrzeit` und ein `Thema` (repräsentiert als `char*`).

Geben Sie eine Definition der Klasse `Termin` an mit einem Default-Konstruktor `Termin()`, einem dreistelligen Konstruktor `Termin(Date, Uhrzeit, char*)` und einer Methode `print()`.

5. Gegeben ist ... [10 Punkte]

die folgende Template-Deklaration:

```
template <class T>
class arr
{ protected: T* _inhalt;
             int _elementanzahl;

  public:    arr(int anz=10) : _elementanzahl(anz), _inhalt(new T[anz]) {}
             arr(const arr<T>&);                    // Kopie
             ~arr();                                // Destruktion
             arr<T>& operator= (const arr<T>&);     // Wertzuweisung
             T& operator [] (int i) {return _inhalt[i];}
             int elementanzahl() const {return _elementanzahl;}
};
```

Definieren Sie die drei fehlenden Elemente der Schablonen-Klasse `arr` für Kopie, Destruktion und Wertzuweisung außerhalb der Klassendefinition. Denken Sie bei der Wertzuweisung an die notwendige Speicherfreigabe.