Name: Matrikelnummer:

# Probeklausur: PRGIII MD

**Aufgabe 1 (20 Punkte)**
Folgende C++ Programme enthalten Fehler, die vom Compiler entdeckt werden. Erklären Sie den Fehler und machen Sie einen Korrekturvorschlag.

**a)**
```
#include <iostream>
int f(){int x=42;}

int main(){
  int y = f();
  int& x = f();
  x=y
}
```

**b)**
```
int main(int argc,char** argv){
  int* pX = new int;
  char* cs [*pX];
  cs[4]   = argv[2];
  *pX     = cs[0];
}
```

**c)**
```
#include <iostream>
class Date{
    int year;int month; int day;
    Date(int year,int month,int day):
        year(year),month(month),day(day){};
};

int main(){
  Date d=Date(2004,1,26);;
  std::cout << "starting main" << std::endl;
}
```

**d)**
```
#include <string>
#include <iostream>
using namespace std;

class Person{
 public:
   string name; string vorname;
   Person(string name,string vorname)
    :name(name),vorname(vorname){}
   string getFullName(){return vorname+" "+name;}
```

```
};

int main(){
  char* n ="Zerlett";
  string v ="Helmut";
  Person p = new Person(n,v);
  cout << p.getFullName()<<endl;;
  cout << p.vorname<<endl;;
}
```

**Aufgabe 2 (24 Punkte)**
Rechnen Sie folgende Programme auf dem Papier und geben Sie die Ausgabe auf dem Bildschirm an. Erklären Sie kurz, wie es zu der Ausgabe kommt.

**a)**

```
#include <iostream>
int main(){
  int* x = new int[2];
  *(x+1) = 42;
  int y  = 7;
  *x     = y+*(x+1);
  std::cout << y << std::endl;
  std::cout << x[0]   << std::endl;
  std::cout << *(x+1)   << std::endl;
}
```

**b)**

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

class C1 {
public:
  string getDescription(){return "C1";}
};

class C2: public C1 {
public:
  string getDescription(){return "C2";}
};

int main(){
    C1* c1 = new C1();
    C1* c2 = new C2();

    cout << c1->getDescription()<< endl;
```

```
20      cout << c2->getDescription()<< endl;
21  }
```

**c)**

```
1   #include <iostream>
2   using namespace std;
3
4   template <class At>
5   class Box{
6   public:
7     At content;
8     Box<At>(){};
9     Box<At>(At x):content(x){};
10  };
11
12  template <typename At>
13  void fB(Box<At> b){
14    At x = 2 * b.content;
15    b = Box<At>(x);
16    b.content=42;
17  }
18
19  int main(){
20    Box<int> b(17);
21    fB(b);
22    cout << b.content << endl;
23    Box<int>& c = b;
24    fB(c);
25    cout << b.content << endl;
26  }
```

**Aufgabe 3 (28 Punkte)**
Schreiben Sie folgende generischen Funktionen für die STL:

**a)**

```
1   template <typename Iterator,typename ElementType>
2   void drop(Iterator begin,Iterator end,int i
3           ,vector<ResultType>& result);
```

`drop` soll alle Elemente des Iteratorbereichs ab dem i-ten Element in den Vektor `result` einfügen.

**b)** Schreiben Sie zwei Beispielanwendung der obigen Funktion `drop`: einmal auf einem Iteratorbereich für eine Reihung einmal für den Iteratorbereich auf einem Vektorobjekt.

**c)**

```
template <typename Iterator, typename UnaryFunction>
void myForEach(Iterator begin,Iterator end,UnaryFunction f);
```

`myForEach` soll die gleiche Funktionalität wie die STL Funktion `for_each` haben. Natürlich dürfen Sie die Funktion `for_each` aus der STL nicht verwenden.

**Aufgabe 4 (28 Punkte)**
Schreiben Sie eine generische Klasse für die Darstellung von Abbildungen. Die Klasse soll generisch über den Schlüsseltyp und dem Werttyp der in der Klasse abgespeicherten Paare stehen.

Die Klasse soll folgende zwei Funktionen enthalten:

- void add(Key k,Value v)
- Value lookUp(Key k)

Benutzen Sie hierzu intern die Klasse `vector` aus der STL und schreiben Sie sich eine Hilfsklasse zur Speicherung von Paaren.

## Benotung

| **Aufgabe**: | 1 | 2 | 3 | 4 | | Summe | | Übungen | Gesamt |
|---|---|---|---|---|---|---|---|---|---|
| **Punkte**: | 20 | 24 | 28 | 28 | | 100 | | 10 | |
| **erreicht**: | | | | | | | | | |