Name: Matrikelnummer:

Klausur: Programmieren II

Diplomstudiengang Allgemeine Informatik

SS07

Erlaubte Hilfsmittel: dokumentenechter Stift.

Lösung ist auf den Klausurbögen anzufertigen. (eventuell Rückseiten nehmen)

Bitte legen Sie einen Lichtbildausweis und den Studentenausweis auf den Tisch.

Bearbeitungszeit: 90 Minuten

Unterschrift

Benotung

Aufgabe:	1	2	3	4	5		Gesamt	Note
Punkte:	16	16	24	24	20		100	
erreicht:								

Aufgabe 1 Folgende C++ Programme enthalten Fehler, die vom Compiler entdeckt werden. Erklären Sie den Fehler.

a)

```
#include <iostream>
class A{
  private:
    int x;
  public:
    A(int x):x(x){};
    int getX(){return x;}
};
int main(){
  A a;
  std::cout<<a.getX()<<std::endl;
}
```

b)

```
int f(int& x){x=x+2;return x;}
int main(){f(4);return 0;}
```

c)

```
#include <string>
template <typename a,typename b>
class P{
 public:
  a fst;
  b snd;
  P(a x,b y):fst(x),snd(y){}
  void swap(){b x=snd;snd=fst;fst=x;}
};

int main(){
  P<std::string,std::string>* p1
   = new P<std::string,std::string>("hallo","welt");
  P<int,std::string>* p2 = new P<int,std::string>(17,"4");
  p2->swap();
  p1->swap();
  return 0;
}
```

d)

```
#include <iostream>
class A1 {
 public:
  virtual std::string toString()=0;
  virtual int getAnswer(){return 42;}
};

int main(){
  A1* a1 =  new A1();
  std::cout<<a1->getAnswer()<<std::endl;
  return 0;
}
```

Aufgabe 2 Rechnen Sie folgendes Programm 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 f1(int x){
  int result = 0;
  for (int i=0;i<=x;i++){
    if (i==3) throw result;
    if (x<0) throw "negatives Argument in f1";
    result=result+i;
  }
  return result;
}

int f2(int i){
  try {
    return f1(i);
  }catch (int i){
    return i;
  }
}
int main(){
  std::cout<<f2(12)<<std::endl;
  return 0;
}
```

b)

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

class GeometricObject{
public:
  int hoehe;
  int breite;
  GeometricObject(int h,int b):hoehe(h),breite(b){}
  virtual float flaeche(){return hoehe*breite;}
  float umfang(){return 2*(hoehe+breite);}
};

class Quadrat:public GeometricObject{
public:
  Quadrat(int h):GeometricObject(h,h){}
};

class Kreis:public Quadrat{
public:
  Kreis(int h):Quadrat(h){}
  virtual float flaeche(){return 3.14*breite*hoehe;}
  virtual float umfang() {return 3.14*breite;}
};

int main(){
  Quadrat          geo0 = Quadrat(10);
  GeometricObject& geo1 = geo0;
  GeometricObject  geo2 = Kreis(5);
  GeometricObject* geo3 = &geo2;
  geo0.breite = 20;
  cout<< geo1.flaeche()  <<endl;
  cout<< geo1.umfang()   <<endl;
  cout<< geo2.flaeche()  <<endl;
  cout<< geo2.umfang()   <<endl;
  cout<< geo3->flaeche() <<endl;
  cout<< geo3->umfang()  <<endl;
  return 0;
}
```

Aufgabe 3 Schreiben Sie im Stil der Standard Template Library folgende Funktionen, die das Iterator-Konzept benutzen sollen. Schreiben Sie jeweils einen Test, in dem Sie die Funktion aufrufen.

a)
```
template <typename Iterator>
void printAll(Iterator anfang, Iterator ende);
```
Die Funktion soll alle Elemente des Iterationsbereichs auf die Konsole ausdrucken.

b)
```
template <typename Iterator, typename ElementType>
void wendeAn(Iterator anfang,Iterator ende, void f(ElementType));
```
Die Funktion soll die übergebene Funktion `f` auf alle Elemente, über die Iteriert wird, anwenden.

c) Lösen Sie jetzt noch einmal Aufgabenteil a) mit Hilfe der Funktion aus Aufgabenteil b).

Aufgabe 4 Gegeben sei folgende generische Klasse, die einen Binärbaum realisiert:

```
1 template <typename ET>
2 class Tree{
3  public:
4   ET x;
5   Tree<ET>* leftChild;
6   Tree<ET>* rightChild;
7   Tree(ET x,Tree<ET>* leftChild=0,Tree<ET>* rightChild=0)
8      :leftChild(leftChild),x(x),rightChild(rightChild){};
9 };
```

Ergänzen sie die Klasse um folgende Methoden:

a) `bool isLeaf();`
die genau dann wahr ergibt, wenn die Felder `leftChild` und `righChild` beide einen Nullzeiger haben.

b) `int count();`
die die Anzahl der Knoten des Baumes zurückgibt.

c) Schreiben Sie einen geeigneten Destruktor für die Klasse `Tree`.

Aufgabe 5 Erklären Sie in kurzen Worten die Begriffe den Unterscheid zwischen Überschreiben und Überladen von Methoden. Erläutern Sie in diesem Zusammenhang den Unterschied zwischen dynamischer und statischen Auflösung eines Methodenaufrufs.