

Name	Matr. no.	

Exam Programming III - TI3

SS 12

Prof Dr K. Baer

Aids: 1 sheet DIN A4, labelled on both sides.

Processing time: 90 min.

- 1. Please enter your name and matriculation number first!
- 2. Check that the task sheets are complete.
- 3. The exam consists of 6 tasks. Get a brief overview of the tasks and start with the task that is most likely to give you a sense of achievement.
- 4. Read the task carefully before solving it!
- 5. Use the space provided on the task sheets to answer the questions.
- Write legibly. Illegible parts will be awarded 0 points!

Good luck!

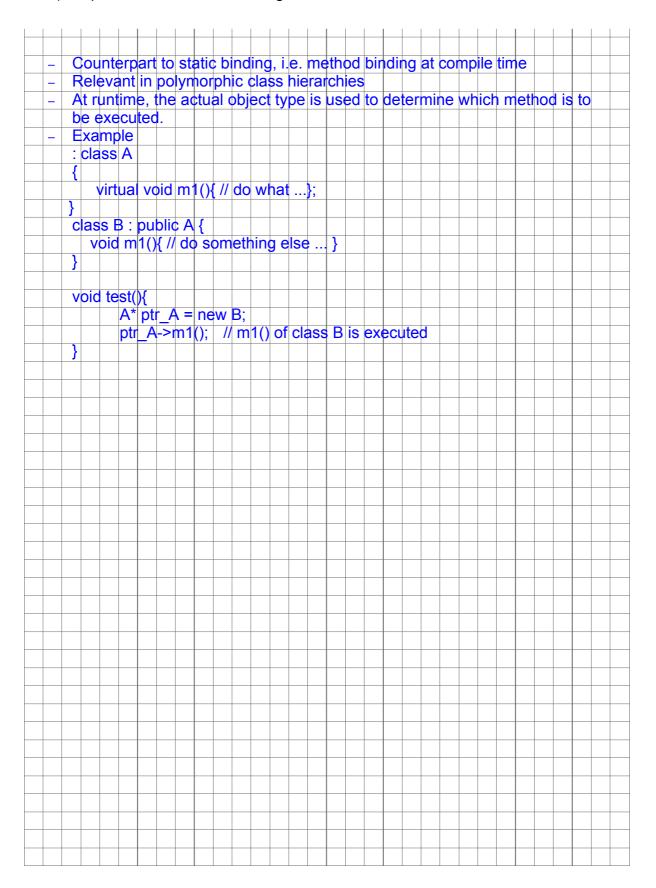
Task	1	2	3	4	5	6	Total
Points	20	6	14	20	6	24	90
achieved							

Task 1 (20 points = 2+11+2+2+1+2)

Given the following implementation:

```
class Component
{ protected:
       string name;
public:
       Component(){ name = "???"; cout << "Component created" << endl; }
       ~Component(){ cout << "Component destroyed" <<
       endl;} void showName(){ cout << "Name: " << name <<
       endl; }
};
class A : public Bauteil {
public:
       A() { name = "A"; cout << name << " created" << endl; }
       ~A(){ cout << name << " destroyed" << endl; }
};
class B : public Bauteil {
public:
       B() { name = "B"; cout << name << " created" << endl; }
       ~B(){ cout << name << " destroyed" << endl; }
       void showName(){ cout << "B-Parts" << endl; Component::showName(); }</pre>
};
class Assembly {
private:
       set<component*> group;
       typedef set<component*>::iterator
Iter; public:
       Assembly(){ cout << "Assembly created" << endl; }
       assembly(const assembly& other){ cout << "assembly copied" << endl; for(Iter
             i=other.group.begin(); i != other.group.end(); ++i){
                    group.insert(*i);
             }
       virtual ~assembly(){ cout << "Assembly destroyed" << endl; } void
       hinzufuegen(Bauteil* teil){ gruppe.insert(teil); }
       void remove( component* part){
       group.erase(part); } void-showElements(){------
             cout << endl << "
             cout << "\nAssembly group populated with: " <<</pre>
             endl; for(Iter i = group.begin(); i != group.end();
             ++i){}
                    (*i)->showName();-----
             cout << "
                                                " << endl;
       }
};
```

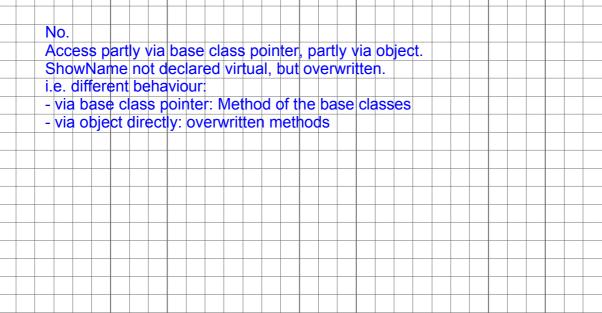
a) Explain the term "late bonding"!



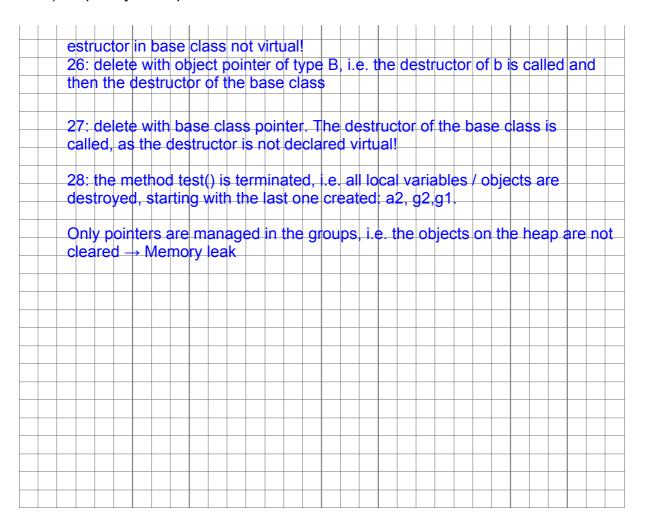
b) What does the following test programme output?
For clarification, please enter the **line number** for each output line in the test() method that generates this output.

```
8Assembly group created
                                       10Component
 1
     #include "bauteil.h"
2
                                       created A
                                       created
 3
     void display(assembly group)
                                       {11Component created
 4
                                      B created
      group.showElements();
 5
                                12
                                      Component generated
     }
                                                                    \times 0.5
                                                                               2
 6
                                       B created
 7
     void test(){
                                18
                                      Module copied
                                                                                1
 8
      Assembly group group1;
 9
                                18
                                                                                1
10
       A* a1 = new A();
                                     Assembly equipped with:
11
                                     Name: A
       B* b1 = new B();
    Component* = new B();
12
                                     Name: B
             b2
13
                                     Name: B
                                     -----
14
      group1.add(a1);
15
      group1.add(b1);
                                18
                                       Assembly destroyed
                                                                                1
16
      group1.add(b2);
                                20
                                                                                1
                                     B-parts
17
                                      Name: B
18
      display(group1);
                                22
                                       Module copied
                                                                                1
19
                                                                                0,5
                                23
                                      | Component | generated
20
      b1->showName();
                                       A created
21
22
      Assembly group group2 = group1; 25
                                                                                1
23
                                     Assembly equipped with:
24
      group2.add(&a2);
                                     Name: A
25
                                     Name: | A
      group2.showElements();
26
      delete b1;
                                     Name: B
27
      delete b2;
                                     Name: B
28
                                            ------
29
                                26
                                       B destroyed
                                                                                0,5
30
     int main(){
                                       Component destroyed
31
                                       Component destroyed
     test();
                                27
                                                                                0,5
32
      cin.sync();cin.get();
                                28
                                       A destroyed
                                                                                0,5
33
                                       Component destroyed
                                       Assembly destroyed
                                                                                0,5
                                       Assembly destroyed
                                                                                0,5
```

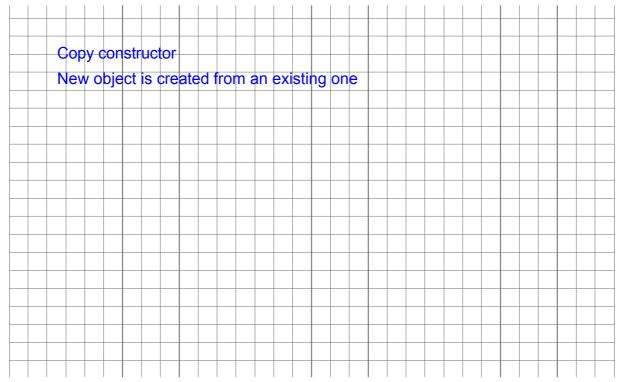
c) Component b1 is output at different locations. Does it always appear the same? Give reasons for your answer.



d) Explain your expenditure on lines 26-28.



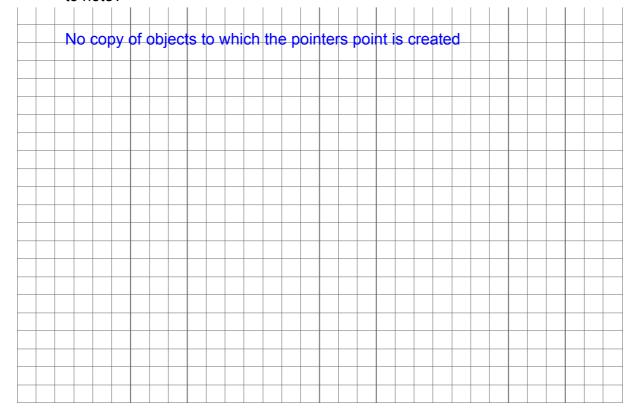
e) Explain in detail which functions are called in line 22!



f) What is required to implement the function

Assembly(const assembly& other)

to note?



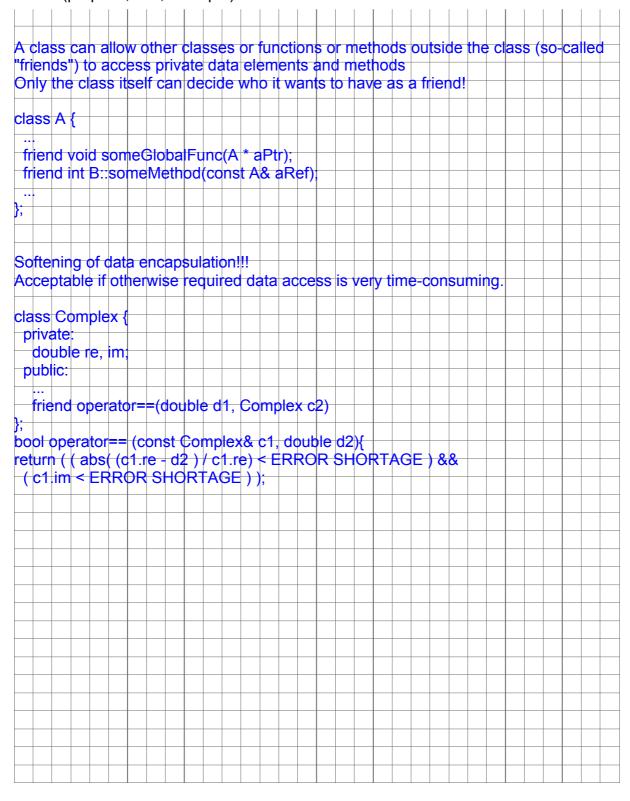
Task 2 (6 points)

What is the sensible approach to operator overloading? Explain - using the example of a fraction class - which problems typically occur and how best to deal with them. For example, in the case of the class Fraction, how do you ensure that the implementation fulfils the commutative law?



Task 3 (14 points=7+7)

a) Explain the friend concept in C++! (purpose, use, example)



b) What is a binder in the STL? (purpose, use, example)

						ı	-		_										-	-	1					-			_		
Wh																															
Pu	rp																									rs					
		F	or	t	WC	-v	al	ue	d	fυ	ınc	tc	rs	; ,	or	ne	01	E t	the	e (ppe	era	and	ds							
		b	ou	nd	h	У	bi	nc	er	s.	T	'hi	S	m∈	ear	ıs	tł	hat	-	þn.	ГУ	2	-V	al [·]	ue	d	£u	nc	to	rs	_
\Box															f	r	b	lno	liı	hg	t.	he	1:	st	0	þе	ra	nd	a	nd	_
						t														-						Ė					
											-																		_		
te	mp	la	te	<c< td=""><td>la</td><td>ISS</td><td></td><td>ne</td><td>ra</td><td>ti</td><td>or</td><td>1 .</td><td>c]</td><td>as</td><td>ss</td><td>T)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></c<>	la	ISS		ne	ra	ti	or	1 .	c]	as	ss	T)															
cl	as	S	Βi	nd	_ r	- (1																							
pu						<u> </u>																									
μu	DΙ	TC	in	40	~ 1	CO	ne	+	Or	~ r	· a +	· i c	n s			~	n	×+	Ιт.	c. +	- \			\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	_ \	\ _{5.7}	/+	\	ſ	ι	
		—Þ	111	1	- 1	er	113	ر -	1/1	/ -	at	+	, п	נ (זר	\)113) L		r .	- /		~~ 1) (^:) 	, У 		<i>)</i>	l 1	J	
					ob	er	аl	10	()	((10:	St		Ċα	-X)	(100	ısı	-	١.	re	ьu.	<u>rn</u>	_0	Р (Χ,	Y) ;	-}		
pr	lV							, ,																							
						ре		Ll	or		p;																				
			on	st	_ T	У	;_																	_		_			_		
};									_																				_		_
Εx	am	рl	e:																												
in tr tr	t	f[]	=	{	3,	1	,	4,	-	,	7,	(),	8	2	2,	6	}	;											
tr	an	sf	or	m (f,	f	+	. 9	,	f,	k	oir	nd2	nc	d (p	οlι	ıs<	<in< td=""><td>nt:</td><td>> (</td><td>,</td><td>1</td><td>))</td><td>;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></in<>	nt:	> (,	1))	;							
tr	an	sf	or	m (f,	f	+	. 9	,	f,		st	re	ear	1 -	Lte	era	ato	br.	<i< td=""><td>nt</td><td>> (</td><td>CO.</td><td>ut</td><td>,</td><td>**</td><td>")</td><td>,</td><td></td><td></td><td></td></i<>	nt	> (CO.	ut	,	**	")	,			
					<u> </u>				<u> </u>						-							<u> </u>			_		<u>'</u>	_			
															_																
																															_
																															_
																															_
\vdash																													_		_
																															_
																															_
																															_
																															_
																															_
\vdash						_																		_		_			_		_

Task 4 (20 points)

A matrix can be understood as an array of fields.

When using the STL, you can easily develop a template class for a matrix by creating a template class

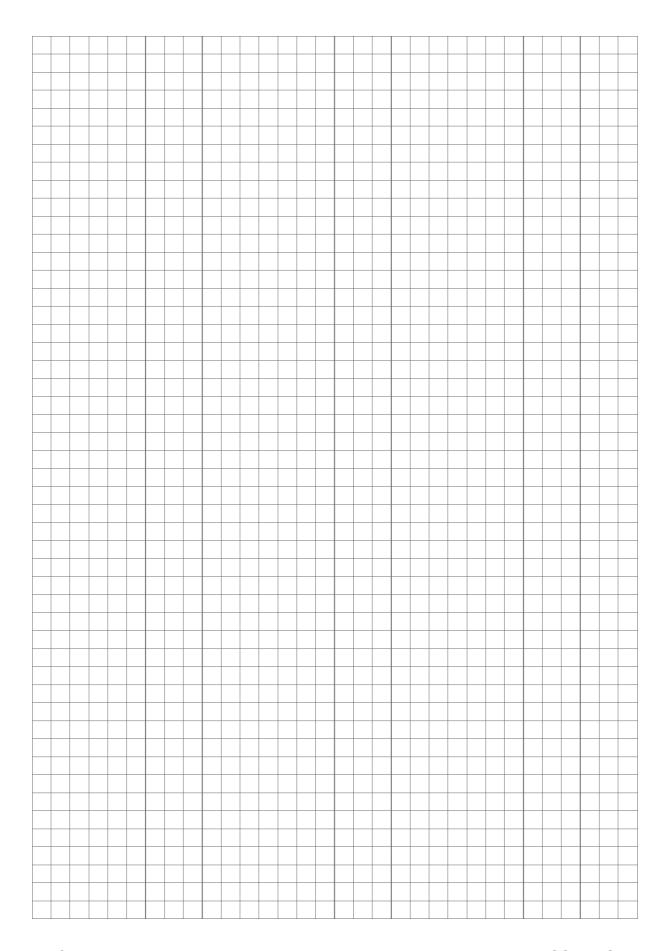
matrix, which is publicly derived from vector < vector < T>>. Develop such a template class with the following public methods:

- a 2-digit constructor for specifying the matrix dimension
- the Rows () and Columns () functions for determining the dimensions
- an init() function that allows the matrix elements to be initialised with a specified value
- a function I () that returns the unit matrix (0 everywhere, only diagonal occupied by 1)
- the output operator operator <<, with which the matrix is output in the form of the following unit matrix (i.e., row no.: values separated by spaces)

0: 1 0 0 0 1: 0 1 0 0 2: 0 0 1 0



				re	etu	rn	*th	is;																			
														}/	/ 3												
pro	ote	cted	4.																								
P		si	- -	typ) <u>e</u>	rOM	18.																				
		ci-	_	typ) C	201	ım	ne						11	2												
		312	4 -	Lyp	י שי	LUI	ulli	113	,					11													
2-1	,		N /																								
},//	\ \cdot	ass	IVI	atri	X																						
ter	npl	ate	<cl< td=""><td>ass</td><td>s T</td><td>></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></cl<>	ass	s T	>																					
inl	ine	ost	rea	ama	% с	pe	rat	or≤	<<(os	tre	am	&	s, c	cor	ıst	Ma	atri	χ<	Γ>	& r	n){					
		ty	pec	lef	tyŗ	er	an	ne	Ma	ıtri	χ<	Γ>:	:si	ze	ty	pe	siz	ze_	tyr	e;							
		fo	r(si	ze	tv	pe	i =	0:	i <	m	.R	ow	s()	: +	+i)	{								\exists			
	+	+	1	9	,, <<	6	ndl		: i	<<	W	i	- ()	,	-,												
	+		+	fr	or(174	ا د ا	VD	، i د	= ()· i	, <	m (\mathbb{C}	lun	าทร	:():	+-	٠i١					\dashv			
	+	+	\vdash	ze_ s fo) I	اے،ر ا	ا_'	ام تر	o j mli	, 1717	',] '	- "	11. 11.		uii	n ic	, (),		J <i>)</i>					-		-	
		+	-				3	`	11[1	וחו			,											-			
		}																									
				en																							
		re	tur	n s	,																						
												}//	6														
	_																										
	\perp		-																								
																								_]			
	+		+																					-			-
	+		-																								-
	-		-										_											-			
	\perp		-																								
	\top																										
	+		+																					\dashv			
	-		+										-													-	

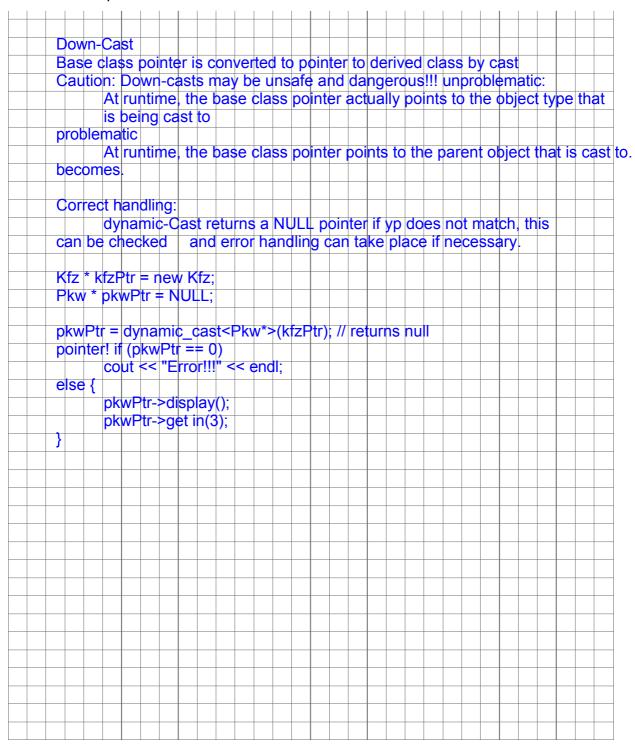


Task 5 (6 points)

Explain what is meant by a downcast.

Describe the situations in which downcasts are problematic. Explain how to handle them correctly.

Give an example.



Task 6 (24 points = 2+1+6+4+11)

Develop a "PIzOrtMapper" class that supports the mapping of postcodes and cities.

Given is a text file according to the sample below, in which an assignment of postcode to city is specified for each line:

```
89075 Ulm
89076 Ulm
89077 Ulm
89077 Ulm-
     Soeflingen
89078 Ulm
89079 Goegglingen
89079 Ulm
89079 Ulm-Danube
     Valley
89079 Ulm-Eggingen
89079 Ulm-Einsingen
89079 Ulm-
     Goegglingen
89079 Ulm-
     Unterweiler
89079 Ulm-
     Wiblingen
89080 Ulm
89081 Ulm
89081 Ulm-Jungingen
89081 Ulm-Lehr
89081 Ulm-
     Maehringen
89081 Ulm-
     Seligweiler
89081 Ulm-
   Soeflingen
89081 Ulm-Ermingen
89081 Ulm-Jungingen
```

This shows that a city can have several postcodes (e.g. Ulm). However, a postcode can also be valid for several towns (e.g. 89079)

a) Explain the advantages of a MultiMap.What elements does a map / multimap consist of?

Efficient access via key (logarithmically) sorted several entries / keys possible pair<const Key, T>

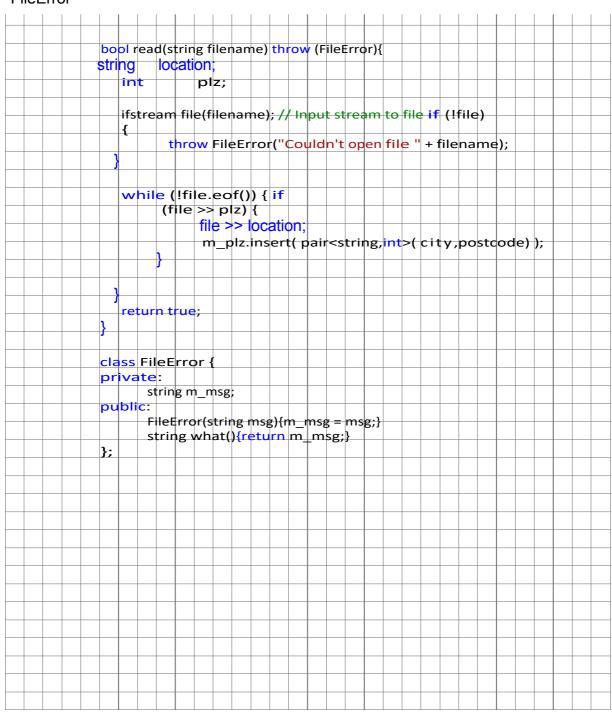
b) Develop the "PlzOrtMapper" class. It should support the following functions:

bool einlesen(string dateiname) throw (FileError); int zaehleVorkommen(string ort); string listPlzOrt(...); First of all, think about a suitable data structure with which you can most easily manage postcode/location pairings within your class. If the file cannot be opened, the "read in" method should generate a corresponding "FileError" error object for exception handling. The file name should be taken from the error object at the calling point. The method zaehleVorkommen() returns the number of postcode/location pairings for a given location. The method listPlzOrt() should return postcode/city pairings as a string. It should be possible to call the function in such a way that all pairings with the exact place name. all pairings in which the city name contains the given search string, all pairings that belong to a postcode are listed. Notes on the STL: Multimap offers an element function "equal range", which can be used to determine the limits of an area in the map in which the key has a certain pair<iterator, iterator> equal range (donst key type& x); The distance between two iterators can be determined using the global function distance size t = distance(iterator1, iterator2); The global function find if can be used to search for values in a container that fulfil a specific condition (a predicate). To do this, the function requires the area in the container to be searched and the comparison function. It returns the position of the first element found (or *last*, if nothing was found). template <class InputIterator, class Predicate> InputIterator find if (InputIterator first, InputIterator last, Predidate pred);

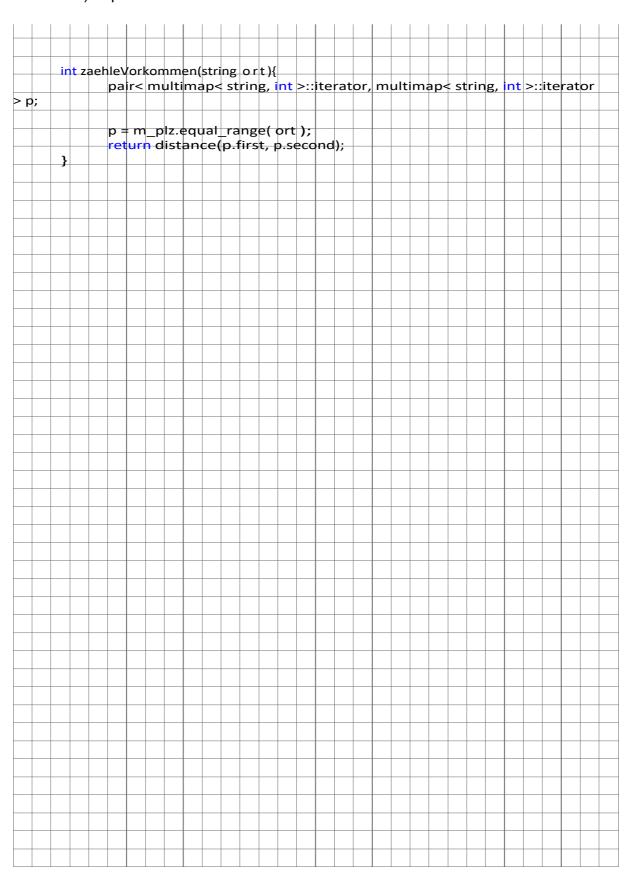
b1) Declare the internal **data structure** for recording and managing postcode/city pairings.



b2) Implement the method **read in** together with the error class "FileError"



$\ensuremath{\texttt{b3}}$) Implement the method $\ensuremath{\textbf{zaehleVorkommen}}$



b4) Implement the method listPlzOrt.

Assistance:

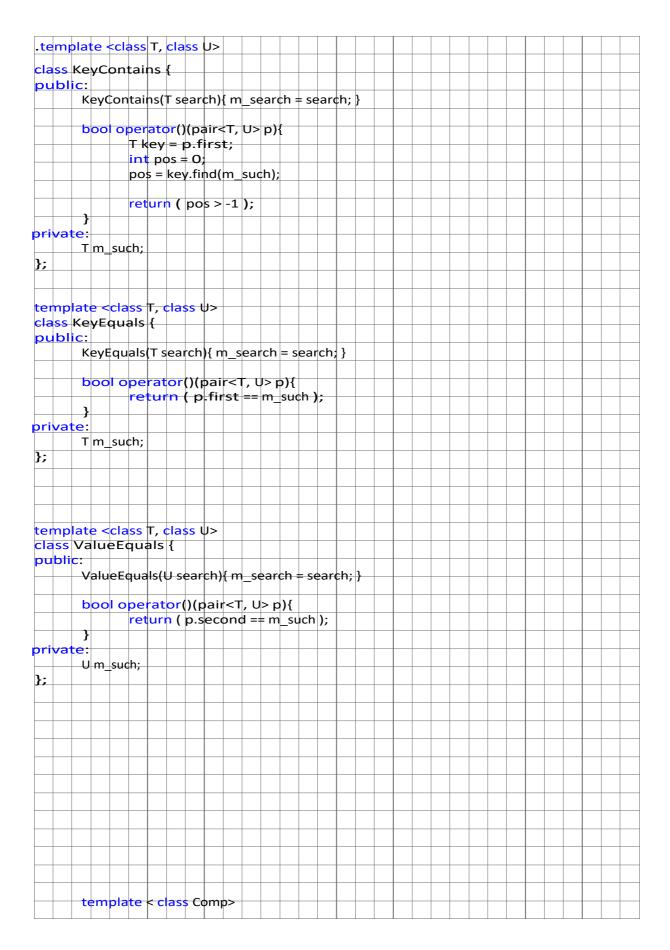
- You can map the different searches using different functors
- Once the search is for the postcode (\rightarrow int), the other times for the city (\rightarrow string). If you implement listPlzOrt as a template, you can save the corresponding implementation effort.
- The usual output operator << can be used to write to stringstream objects as on the console. Stringstream objects have a methdoe str() for conversion to a string.
- Strings offer the function

```
size_t find ( const string& str, size_t pos = 0 ) const;
which returns the position of the first occurrence of the search
```

string. A test programme should provide output according to the

following pattern:

```
List postcode(s) for location: Ulm
89070 Ulm
89073 Ulm
89074 Ulm
89075 Ulm
... (etc.)
List postcodes for places containing the name element: Soef 89077
Ulm-Soeflingen
89081 Ulm-Soeflingen
List of all places belonging to the same postcode: 89079
89079 Goegglingen
89079 Ulm
89079 Ulm-Donautal
89079 Ulm-Eggingen
89079 Ulm-Einsingen
89079 Ulm-Goegglingen
89079 Ulm-Unterweiler
89079 Ulm-Wiblingen
```



```
string listPlzOrt( Comp cmpFunc){
  stringstream buffer;
  multimap< T, U >::iterator i = m_plz.begin(); while
  ( i != m_plz.end() ){
      i = find_if(i, m_plz.end(), cmpFunc);
       if ( i != m_plz.end() ){
             buffer << i->second << "
                                       " << i+>first << endl;
             ++i;
       }
  return buffer.str();
}
```

