Programmierkurs 2

Aufgaben für die Vorbereitung auf die Klausur im
Wintersemester 2016/17

Inhaltsverzeichnis

Versionshistorie

- 0.1.9
  - Added „Funktionszeiger“
  - Added „Funktionszeiger 2“
- 0.1.8
  - Polished some code designs
- 0.1.7
  - Added „Malloc & free“
- 0.1.6
  - Added „Libraries“
- 0.1.5
  - Added „Zeigerarithmetik & Arrays“
  - Added „Zeigerarithmetik & Arrays 2“
  - Added „Zeigerarithmetik & Arrays 3“
- 0.1.4
  - Added TODOs
- 0.1.3
  - Design overhaul
- 0.1.2
  - Added „Struts and Unions“
  - Added „Struts and Unions 2“
- 0.1.1
  - Added „Startpunkt“
  - Added „Makefile“
- 0.1.0
  - Added „GNU Free Documentation License“
- 0.0.1
  - First document design

C

Startpunkt

Was hiervon sind valide Deklarationen der Main Methode? (ankreuzen)

int main(int argc, char *argv[]) { return 0; }	□
int main(){ return 0; }	□
float main(){ return 0; }	□
void main() { return 0; }	□
int main(int argc, char **argv) { return 0; }	□
int main(void) { return 0; }	□
char* main() { return 0; }	□

Makefile

Situation: Zwei Quelldateien main.c und summe.c sowie die Headerdatei summe.h, die main.c ruft die Methode `int make_sum(int a, int b)`.

Schreiben Sie ein makefile, welches die Dateien kompiliert und eine ausführbare Datei „main“ erstellt. Beachten Sie dabei, dass die Anweisungen die Abhängigkeiten der Quelldateien beachten(Bonus wenn beim Kompilieren der richtige C-Standard angegeben wird)

main.c

```
#include <stdio.h>
#include "summe.h"

int main(void)
{
	printf("Summe von 9 und 21 ist %d\n", make_sum(9,21));
}
```

summe.c

```
int make_sum(int a, int b)
{
	return a+b;
}
```

summe.h

```
int make_sum(int a, int b);
```

makefile

```

```

Structs und Unions

Sie sehen das Programm auf der rechten Seite, es ist laut C11 Standard valide und kompiliert ohne Fehler.
Wie sieht die Ausgabe des Programms aus?

```
s.a =
s.b =
u.a =
u.b =
```

```c
#include <stdio.h>

struct teststruct {
	int a;
	char b;
};

union testunion {
	int a;
	char b;
};

int main()
{
	/*	Auszug aus der ASCII Tabelle
		65 = A
		66 = B*/
	struct teststruct s;
	s.a = 65;
	s.b = 66;

	printf("s.a = %d\n", s.a);
	printf("s.b = %c\n", s.b);

	union testunion u;
	u.a = 65;
	u.b = 66;

	printf("u.a = %d\n", u.a);
	printf("u.b = %c\n", u.b);

	return 0;
}
```

Structs und Unions 2

Welche der drei unteren Aussagen trifft zu?

Die Struktur **s** ist…

kleiner als das Union **u**	□
gleich groß wie das Union **u**	□
größer als das Union **u**	□

Typedef

Ergänzen Sie den nebenstehenden Programmcode so, dass ein neuer Typ definiert wird mit Namen **point**, der auf die Struktur **point_s** zeigt, sodass der untere Code nach C11 Standard valide ist und kompiliert.

```c
#include <stdio.h>

struct point_s {
	int x;
	char y;
};

//hier kommt die Typdefinition hin:


int main()
{
	struct point_s p1 = { 5, 7 };
	point p2 = { 3, 2 };

	printf("p1.x = %d, p1.y = %d\n", p1.x, p1.y);
	printf("p2.x = %d, p2.y = %d\n", p2.x, p2.y);

	return 0;
}
```

Zeigerarithmetik & Arrays

Was sind laut C11 Standard valide Deklaration für ein Array aus Integer Werten? Wie groß ist das Array(nichts angeben, falls es sich um eine nicht valide Deklaration handelt)

Deklaration	valide?	Maximale Anzahl der enthaltenen Elemente?
int arr1[] = { 2 };	☐	
int arr2[];	☐	
int arr3[1];	☐	
int arr4[2] = { 6 };	☐	

Zeigerarithmetik & Arrays 2

Ergänzen Sie den unteren Programmcode so, dass die Methode **void uppercase(?)** einen Zeiger erwartet, dessen Wert überprüft wie im Programmcode angegeben und ihn ggf. auf einen neuen Wert setzt. Beim Aufruf der Methode **void uppercase(?)** soll ein Zeiger auf den **char c1** übergeben werden, nicht **c1** selber.

```
#include <stdio.h>

void uppercase(               )
{
	char tmp =               ;//Wert von Übergabeparameter der Variable tmp zuweisen

	//Prüfe ob der Wert im lowercase Bereich liegt
	if(tmp >= 'a' && tmp <= 'z')
	{
		tmp = tmp - 'a' + 'A';

		      = tmp; //Weise dem Speicher, auf den der Zeiger zeigt, den Wert von tmp zu
	}
}

int main()
{
	char c1 = 'a';

	printf("c1 = '%c'\n", c1);

	printf("making c1 uppercase..\n");
	uppercase(               );

	printf("c1 = '%c'\n", c1);

	return 0;
}
```

Die Ausgabe des korrekt vervollständigten Programms würde dann so aussehen

```
c1 = 'a'
making c1 uppercase..
c1 = 'A'
```

Zeigerarithmetik & Arrays 3

Vervollständigen Sie die Methode **void uppercase_string(char * str)** so, dass alle Elemente eines übergebenen nullterminierten Chararrays uppercase sind

```
#include <stdio.h>

int work(char* str);
void uppercase_string(char* str);

int main()
{
	char str1[] = { 'h', 'e', 'l', 'l', 'o', '\0' };
	char str2[] = "world";

	work(str1);
	work(str2);

	return 0;
}

int work(char* str)
{
	printf("str = '%s'\n", str);

	printf("making str uppercase..\n");
	uppercase_string(str);

	printf("str = '%s'\n", str);
}

void uppercase_string(char* str)
{
	int i;
	for(i=0;            != '\0'; i++)
	{
		if((            >= 'a') && (            <= 'z'))
		{
			       =      - 'a' + 'A';
		}
	}
}
```

*In C gibt es von Haus aus nicht den Datentyp String, stattdessen wird dieser Datentyp über ein Chararray simuliert, dessen letztes Element ein '\0' ist, sodass man nicht die Länge des Arrays angeben muss. Dieser simulierte String lässt sich einmal über die ganz normale Arraydeklaration definieren, wobei das letzte Element ein '\0' ist(in der Aufgabe Variable **str1**). Der Nullterminator muss nicht am Ende des eigentlichen Arrays stehen, aber dies hat zur Folge, dass Methoden wie z.B. printf() alle Elemente nach dem Nullterminator ignorieren – gleichbedeutend ist das Verhalten undefiniert wenn ein so simulierter String keinen Nullterminator enthält. Weiterhin kann man ein nullterminiertes Chararray auch durch die Deklaration mit doppelten Anführungszeichen erzeugen(in der Aufgabe Variable **str1**), was deutlich einfacher und natürlicher ist, aber komplett gleichbedeutend mit der Arraynotation ist.*

Die Ausgabe des korrekt vervollständigten Programms würde dann so aussehen

```
str = 'hello'
making str uppercase..
str = 'HELLO'
str = 'world'
making str uppercase..
str = 'WORLD'
```

Funktionszeiger

Vervollständigen Sie das untere Programm so, dass die Funktion `int work(int a, ?)` einen `int` und einen Funktionszeiger, welcher als Parameter sowie als Rückgabe einen `int` hat, erwartet. Die Funktion `int work(int a, ?)` ruft den Funktionszeiger auf und gibt den Wert zurück.

Weiterhin soll die `int main()` Funktion zwei Funktionszeiger erstellen, mit denen nachher die Funktion `int work(int a, ?)` aufgerufen wird. Der erste Funktionszeiger `fp1` soll auf die Funktion `int add_two(int a)` verweisen, der zweite `fp2` auf die Funktion `int add_multiply_by_three(int a)`.

```c
#include <stdio.h>

int add_two(int a)
{
	return a+2;
}

int multiply_by_three(int a)
{
	return a*3;
}

int work(int a,                    )
{
	return                 ;
}

int main()
{



	int a = 7;
	printf("work(a, fp1) = %d\n", work(a, fp1));
	printf("work(a, fp2) = %d\n", work(a, fp2));

	return 0;
}
```

Die Ausgabe des korrekt vervollständigten Programms würde dann so aussehen

```
work(a, fp1) = 9
work(a, fp2) = 21
```

Funktionszeiger 2

Was gibt das untere Programm bei Ausführung aus?

```c
#include <stdio.h>

int add(int a, int b)
{
	return a+b;
}

int subtract(int a, int b)
{
	return a-b;
}

int multiply(int a, int b)
{
	return a*b;
}

int divide(int a, int b)
{
	return a/b;
}

int main()
{
	int (* fp1) (int, int) = add;
	int (* fp2) (int, int) = &subtract;
	int (* fp3) (int, int) = *multiply;
	int (* fp4) (int, int) = **divide;

	int a = 7;
	int b = 3;
	printf("fp1 = %d\n", (*fp1) (a, b));
	printf("fp2 = %d\n", fp2(a, b));
	printf("fp3 = %d\n", (*fp3) (a, b));
	printf("fp4 = %d\n", fp4(a, b));

	printf("etwas Action reinbringen..\n");
	fp2 = fp1;
	fp1 = fp3;
	printf("fp1 = %d\n", (*fp1) (a, b));
	printf("fp2 = %d\n", fp2(a, b));
	printf("fp3 = %d\n", (*fp3) (a, b));
	printf("fp4 = %d\n", fp4(a, b));

	return 0;
}
```

Ausgabe:

```
fp1 =
fp2 =
fp3 =
fp4 =
etwas Action reinbringen..
fp1 =
fp2 =
fp3 =
fp4 =
```

Types & sizes

Libraries

Sie haben die Quelldateien **auto.c**, **fahrrad.c** die zu einer Library **fahrzeug.a** hinzugefügt werden sollen via den Anweisungen im makefile und dem Konsolentool „ar"

auto.c

```c
#include <stdio.h>
#include "fahrzeug.h"

Auto new_auto(char* bezeichnung, int ps, double preis)
{
	Auto a = { bezeichnung, ps, preis };
	return a;
}

void print_auto(Auto a)
{
	printf("bezeichnung = %s, ps = %d, preis = %.2f", a.bezeichnung, a.ps, a.preis);
}
```

fahrrad.c

```c
#include <stdio.h>
#include "fahrzeug.h"

Fahrrad new_fahrrad(char* bezeichnung, double preis)
{
	Fahrrad f = { bezeichnung,  preis };
	return f;
}

void print_fahrrad(Fahrrad f)
{
	printf("bezeichnung = %s, preis = %.2f", f.bezeichnung, f.preis);
}
```

fahrzeug.h

```c
#ifndef FAHRZEUG_H_
#define FAHRZEUG_H_

typedef struct {
	char* bezeichnung;
	int ps;
	double preis;
} Auto;

Auto new_auto(char* bezeichnung, int ps, double preis);
void print_auto(Auto a);

typedef struct {
	char* bezeichnung;
	double preis;
} Fahrrad;

Fahrrad new_fahrrad(char* bezeichnung, double preis);
void print_fahrrad(Fahrrad f);

#endif
```

main.c

```c
#include <stdio.h>
#include "fahrzeug.h"

int main()
{
	Auto a = new_auto("Ford Fiesta", 42, 1337.0);
	Fahrrad f = new_fahrrad("Hollandrad", 420.0);

	printf("Auto a: ");
	print_auto(a);
	printf("\n");

	printf("Fahrrad f: ");
	print_fahrrad(f);
	printf("\n");

	return 0;
}
```

makefile

```
```

Bonus:

Warum sind in der Headerdatei **fahrzeug.h** diese Compileranweisungen enthalten?

```c
#ifndef FAHRZEUG_H_
#define FAHRZEUG_H_

[...]

#endif
```

Malloc & free

Vervollständigen Sie das Programm so, dass die Funktion char* create_string(int length, char init) ein Chararray dynamisch mittels der Library stdlib.h erzeugt mit length Elementen, die den Wert von init haben. Zudem soll das Chararray nullterminiert sein.

Weiterhin soll die Methode void delete_string(char* str) ein dynamisch erzeugtes Chararray übergeben bekommen und dieses mittels der Library stdlib.h löschen

```
#include <stdio.h>
#include <stdlib.h>

char* create_string(int length, char init)
{
	char* str =                                         ;

	int i;
	for(i=0;                    ; i++)
	{
		str[i] = init;
	}



	return str;
}

void delete_string(char* str)
{

}

int main()
{
	char* str1 = create_string(5, 'a');
	char* str2 = create_string(9, 'B');

	printf("str1 = %s\n", str1);
	printf("str2 = %s\n", str2);

	delete_string(str1);
	delete_string(str2);

	return 0;
}
```

Die Ausgabe des korrekt vervollständigten Programms würde dann so aussehen

```
str1 = aaaaa
str2 = BBBBBBBBB
```

C++

Classes & inheritance(TODO)
Namespaces(TODO)
Input & output / streams(TODO)
Operator overloading(TODO)
Exceptions(TODO)
Multiple inheritance(TODO), maybe?
Inline functions(TODO), maybe?
Virtual methods(TODO)
Templates(TODO)

C#

Preamble(TODO)

Input & output(TODO)

Classes & inheritance(TODO)

Namespaces(TODO)

Exceptions(TODO)

Observer model(TODO)

Delegates(TODO)

Events(TODO)

GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five

of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.