

C data types and control constructs

Programming Expertise - session 02

Christian Kappel

AG Genetik
Universität Potsdam

18 April 2024

Outline

**C data types
and control
constructs**

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

① Introduction

② Data types

③ Control constructs

④ Preview

⑤ Questions

Introduction

C sessions

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- Introduction to C
- **Data types and control constructs**
- Functions
- Arrays and pointers
- More on pointers, trees, linked lists
- File I/O

Hello world!

C data types and control constructs

Christian
Kappel

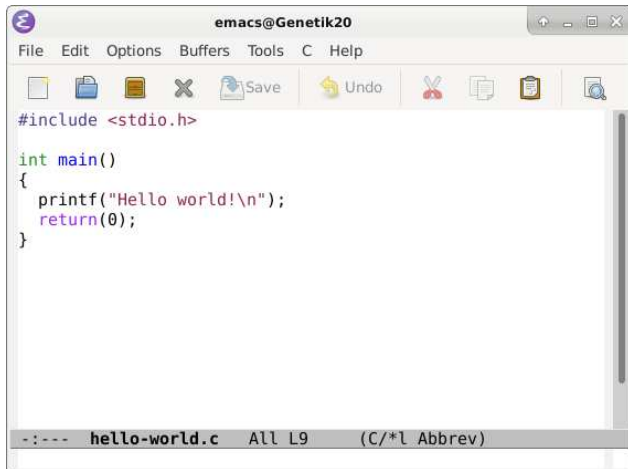
Introduction

Data types

Control
constructs

Preview

Questions



The screenshot shows the Emacs editor window titled "emacs@Genetik20". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The toolbar contains icons for file operations and editing. The code in the buffer is as follows:

```
#include <stdio.h>

int main()
{
    printf("Hello world!\n");
    return(0);
}
```

The status bar at the bottom indicates the file is "hello-world.c", the cursor is at "All L9", and the encoding is "(C/*l Abbrev)".

gcc, the GNU compiler collection

- gcc for Windows: mingw-w64 (<http://mingw-w64.org/doku.php>).
- Online compiler: https://www.tutorialspoint.com/compile_c_online.php.

The screenshot shows a web browser window with the URL `tutorialspoint.com/compile_c_online.php`. The page title is "Online C Compiler - Online". The interface includes a header for "codingground" with the tagline "SIMPLY EASY CODING". Below the header, there's a section for "Compile and Execute C Online (GNU GCC v7.1.1)". The main area is divided into two panes. The left pane, titled "Execute", contains a C program:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, World!\n");
6
7     return 0;
8 }
```

 The right pane, titled "Result", shows the output of the compilation and execution:

```
$gcc -o main *.c
$main
Hello, World!
```

Data types

Variables

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

Variables refer to data storage locations.

Variables have associated data types.

Variable names:

- Letters, digits and underscore (a-zA-Z0-9_); case matters.
- Must start with letter or underscore (not advised).
- C language keywords are excluded.

Variables

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```
1 | #include <stdio.h>
2 |
3 | int main()
4 | {
5 |     int a,b,res; // variables need to be typed
6 |
7 |     a = 2;
8 |     b = 3;
9 |
10 |    res = a * b;
11 |
12 |    printf("Result is: %d\n",res);
13 |
14 |    return(0);
15 | }
```

Primary data types

- Integer, no fractional part.

Type	Bytes	Minimum range
char	1	[-128, +127]
unsigned char	1	[0, 255]
short	2	[-32,768, +32,767]
unsigned short	2	[0, 65,535]
int	2	[-32,768, +32,767]
unsigned int	2	[0, 65,535]
long	4	[-2,147,483,647, +2,147,483,647]
unsigned long	4	[0, 4,294,967,295]
long long	8	[-9,223,372,036,854,775,808, +9,223,372,036,854,775,807]
unsigned long long	8	[0, 18,446,744,073,709,551,615]

- Minimum ranges, varies depending on system.
- Rule: long long \geq long \geq int \geq short \geq char.

Primary data types

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- To get the info for your system:

```

1 | #include <stdio.h>
2 | #include <limits.h>
3 |
4 | int main() {
5 |     printf("Size in bytes: %d\n", sizeof(char));
6 |     printf("CHAR_MIN: %d\n", CHAR_MIN);
7 |     printf("CHAR_MAX: %d\n", CHAR_MAX);
8 |     printf("Size in bytes: %d\n", sizeof(unsigned char));
9 |     printf("UCHAR_MAX: %d\n", UCHAR_MAX);
10 |
11 |     return(0);
12 | }
```

- CHAR, SHRT, INT, LONG, LLONG

Primary data types

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- Floating-point numbers (real numbers).
- Single-, double- and quadruple-precision numbers.

Type	Bytes	Range
float	4	1.2E-38 to 3.4E+38
double	8	2.3E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

Characters and strings

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- A character is single letter or other symbol.
- A string is a sequence of characters (array of characters).
- Characters are stored numerically.

Type casting

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int i=8, j=4, k=3;
6      printf("i / j = %d\n", i/j); // 2
7      printf("i / k = %d\n", i/k); // 2
8
9      float res = (float) i / k;
10     printf("i / k = %f\n", res); // 2.666667
11     res = i / (float) k;
12     printf("i / k = %f\n", res); // 2.666667
13
14     res = (float) (i / k);
15     printf("i / k = %f\n", res); // 2.000000
16
17     return(0);
18 }
```

Control constructs

for loop

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```
1 | #include <stdio.h>
2 |
3 | int main() {
4 |     int i;
5 |
6 |     for(i=0; i<=10; i=i+1)
7 |     {
8 |         printf("%d\n", i);
9 |     }
10 |
11 |     return(0);
12 | }
```


while loop

```

1  #include <stdio.h>
2
3  int main() {
4
5      // Single statement
6      int i = 0;
7      while(i < 10)
8          printf("i=%d\n", i++);
9
10     // Block of statement
11     int j = 0;
12     while(j < 10) {
13         printf("j=%d", ++j);
14         printf("\n");
15     }
16
17     return(0);
18 }
```

while loop

Relational operators:

Operator	Symbol	Precedence
Equal	==	2
Not equal	!=	2
Greater than	>	1
Less than	<	1
Greater than or equal to	>=	1
Less than or equal to	<=	1

while loop

Precedence example:

```

1 | #include <stdio.h>
2 |
3 | int main()
4 | {
5 |     if(4 > 2 == 3 > 1) // Gives warnings, but works
6 |     {
7 |         printf("Both are true.\n");
8 |         printf("4 > 2 gives %d.\n",4 > 2); // 1
9 |         printf("3 > 1 gives %d.\n",4 > 2); // 1
10 |        printf("4 > 2 == 3 > 1 gives %d.\n",4>2==3>1); // 1
11 |    }
12 |    return(0);
13 | }
```

while loop

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

Mathematical operators:

- Unary operators: increment ($++$), decrement ($--$); prefix or postfix mode.
- Binary operators: $+$, $-$, $*$, $/$, $\%$ (modulus).
- Unary precede binary operators. Use parentheses if necessary.

while loop

```
1  | #include <stdio.h>
2  |
3  | int main() {
4  |
5  |     int c;
6  |
7  |     c = getchar();
8  |     while(c != EOF) {
9  |         putchar(c);
10 |         c = getchar();
11 |     }
12 |
13 |     return(0);
14 | }
```

while loop

```
1 | #include <stdio.h>
2 |
3 | int main() {
4 |
5 |     int c;
6 |
7 |     while((c = getchar()) != EOF) {
8 |         putchar(c);
9 |     }
10 |
11 |     return(0);
12 | }
```

while loop

```
1 | #include <stdio.h>
2 |
3 | int main() {
4 |
5 |     int i = 0, j = 0;
6 |     while(i < 10 && j < 10) {
7 |         i = i + 1;
8 |         j = j + 2;
9 |
10 |        printf("i=%d\tj=%d\n", i, j);
11 |    }
12 |
13 |    return(0);
14 | }
```

while loop

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

**Control
constructs**

Preview

Questions

Logical operators:

Operator	Symbol
AND	&&
OR	
NOT	!

- Use parentheses if necessary.

do-while loop

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```

1  #include <stdio.h>
2
3  int main() {
4
5      int i = 0;
6      do {
7          printf("i=%d\n", ++i);
8      } while(i < 10);
9
10     int j = 0;
11     do {
12         printf("j=%d\n", j++);
13     } while(j < 10);
14
15     return(0);
16 }
```

if statement

```
1 | #include <stdio.h>
2 | int i;
3 |
4 | int main() {
5 |     printf("Give an integer please: ");
6 |     scanf("%d", &i);
7 |
8 |     if(i > 0)
9 |         printf("Gives integer is above zero.\n");
10 |    else if(i < 0)
11 |        printf("Given integer is below zero.\n");
12 |    else
13 |        printf("Given integer is zero.\n");
14 |
15 |    return(0);
16 | }
```

Preview

Functions

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```

1  | #include <stdio.h>
2  |
3  | void print_int(int i); // function prototype
4  |
5  | int main()
6  | {
7  |     int i = 23;
8  |     print_int(i);
9  |     return(0);
10 | }
11 |
12 | void print_int(int i) // function definition
13 | {
14 |     printf("Value of i is %d.\n", i);
15 | }
```

Arrays

- Collection of storage locations having the same data type.

```

1 | #include <stdio.h>
2 |
3 | int main()
4 | {
5 |     int a[10];
6 |
7 |     a[0] = 1;
8 |
9 |     printf("%d\n", a[0]);
10 |
11 |     return(0);
12 | }
```

Strings

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- Strings are character arrays.
- Ending with `\0` (ASCII 0).

```
1 | #include <stdio.h>
2 |
3 | int main()
4 | {
5 |     char a[23] = { 'P', 'o', 't', 's', 'd', 'a', 'm', '\0' };
6 |
7 |     printf("%s\n", a); // Potsdam
8 |     printf("%c\n", a[0]); // P
9 |     printf("%d\n", a[0]); // 80
10 |    printf("%d\n", a[7]); // 0
11 |
12 |    return(0);
13 | }
```

Pointers

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```
1 | #include <stdio.h>
2 |
3 | int main() {
4 |     // Initilize variable i
5 |     int i = 99;
6 |     // Set pointer to variable i
7 |     int *pointer_i = &i;
8 |
9 |     // i and *pointer_i refer to value
10 |    printf("i is %d, *pointer_i is %d\n", i, *pointer_i);
11 |    // &i and pointer_i refer to address
12 |    printf("%p, %p\n", &i, pointer_i);
13 |
14 | }
```

Pointers

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- Address-of operator (&).
- Indirection operator (*).
- array is pointer to the first element of array[].

Dynamic memory allocation

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      // Allocate memory for an user defined integer array
7      int size;
8      scanf("%d", &size);
9
10     int *a;
11     a = (int*) malloc(size * sizeof(int));
12
13     free(a)
14
15     return(0);
16 }
```

Dynamic memory allocation

- For dynamic structures like linked lists:

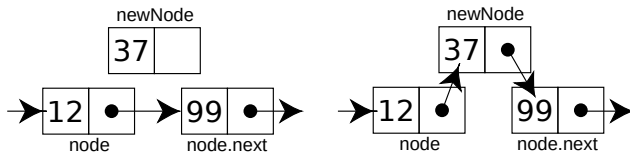


Figure: <https://commons.wikimedia.org/wiki/File:CPT-LinkedLists-addingnode.svg>

Questions

C data types
and control
constructs

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions

- Thank you very much for your attention.
- Please feel free to ask questions.
- You may write to christian.kappel@uni-potsdam.de for any further or upcoming questions.



References I

**C data types
and control
constructs**

Christian
Kappel

Introduction

Data types

Control
constructs

Preview

Questions