Why C?

C was designed in the early 70s as a language specifically for implementing an operating system (Unix)

Why C?

C was designed in the early 70s as a language specifically for implementing an operating system (Unix)

Thus its expertise in a range of low-level, close-to-the machine operations

Why C?

C was designed in the early 70s as a language specifically for implementing an operating system (Unix)

Thus its expertise in a range of low-level, close-to-the machine operations

It was subsequently found to be useful in a wide range of other areas: it started off with a specific application in mind, but turned out to be suitable as a general-purpose language

Why C?

C was designed in the early 70s as a language specifically for implementing an operating system (Unix)

Thus its expertise in a range of low-level, close-to-the machine operations

It was subsequently found to be useful in a wide range of other areas: it started off with a specific application in mind, but turned out to be suitable as a general-purpose language

The original C, designed by Brian Kernighan and Dennis Ritchie (K&R C) was later modified and updated by the standards organisation ANSI

Why C?

The current version of the standard is ANSI C11 (actually ISO standard ISO/IEC 9899:2011, adopted by ANSI), was approved in December 2011

Why C?

The current version of the standard is ANSI C11 (actually ISO standard ISO/IEC 9899:2011, adopted by ANSI), was approved in December 2011

Future revisions are under development

Why C?

The current version of the standard is ANSI C11 (actually ISO standard ISO/IEC 9899:2011, adopted by ANSI), was approved in December 2011

Future revisions are under development

C is very much still a living language

Why C?

Despite its age and origins C is used massively out there in the real world

Why C?

Despite its age and origins C is used massively out there in the real world

In terms of programming jobs, it has been in the top 5 required languages for decades; often in the top 2

Why C?

Despite its age and origins C is used massively out there in the real world

In terms of programming jobs, it has been in the top 5 required languages for decades; often in the top 2

Pretty much anything that is large and “close to the hardware” is programmed in C

Why C?

Despite its age and origins C is used massively out there in the real world

In terms of programming jobs, it has been in the top 5 required languages for decades; often in the top 2

Pretty much anything that is large and “close to the hardware” is programmed in C

It is *very* good when you need low-level, fast programs

Why C?

Despite its age and origins C is used massively out there in the real world

In terms of programming jobs, it has been in the top 5 required languages for decades; often in the top 2

Pretty much anything that is large and “close to the hardware” is programmed in C

It is *very* good when you need low-level, fast programs

For example, real-time controllers of cars, phones, graphics

Why C?

Despite its age and origins C is used massively out there in the real world

In terms of programming jobs, it has been in the top 5 required languages for decades; often in the top 2

Pretty much anything that is large and “close to the hardware” is programmed in C

It is *very* good when you need low-level, fast programs

For example, real-time controllers of cars, phones, graphics

But also in many other situations: it is very flexible and lends itself to many kinds of problem

Why C?

There is a price to pay for this, though

Why C?

There is a price to pay for this, though

C will allow you to do all kinds of stupid things

Why C?

There is a price to pay for this, though

C will allow you to do all kinds of stupid things

Unlike Java, C does not try to protect the programmer from themself

Why C?

There is a price to pay for this, though

C will allow you to do all kinds of stupid things

Unlike Java, C does not try to protect the programmer from themself

If you want to refer to memory that has not been allocated by the system to your program, you can do it

Why C?

There is a price to pay for this, though

C will allow you to do all kinds of stupid things

Unlike Java, C does not try to protect the programmer from themself

If you want to refer to memory that has not been allocated by the system to your program, you can do it

E.g., beyond the end of an array

Why C?

There is a price to pay for this, though

C will allow you to do all kinds of stupid things

Unlike Java, C does not try to protect the programmer from themself

If you want to refer to memory that has not been allocated by the system to your program, you can do it

E.g., beyond the end of an array

The point is that for **experts**, such a thing can be useful, e.g., generating random numbers by reading uninitialised memory

Why C?

But it does not stop non-experts from doing such peculiar things

Why C?

But it does not stop non-experts from doing such peculiar things

And non-experts make many mistakes that can be very dangerous in C

Why C?

But it does not stop non-experts from doing such peculiar things

And non-experts make many mistakes that can be very dangerous in C

In fact, poorly programmed C is one of the major causes of cracks (hacks) of programs and operating systems: many security updates to Windows (amongst others) are to fix just this kind of error

Why C?

But it does not stop non-experts from doing such peculiar things

And non-experts make many mistakes that can be very dangerous in C

In fact, poorly programmed C is one of the major causes of cracks (hacks) of programs and operating systems: many security updates to Windows (amongst others) are to fix just this kind of error

C does not protect the programmer

Why C?

But it does not stop non-experts from doing such peculiar things

And non-experts make many mistakes that can be very dangerous in C

In fact, poorly programmed C is one of the major causes of cracks (hacks) of programs and operating systems: many security updates to Windows (amongst others) are to fix just this kind of error

C does not protect the programmer

Exercise. Read about buffer overflow bugs, use after free bugs, the Heartbleed bug

Why C?

To produce good, safe C programs, you need to be a very good programmer

Why C?

To produce good, safe C programs, you need to be a very good programmer

To produce safe Java programs you don't need to be a good Java programmer

Why C?

To produce good, safe C programs, you need to be a very good programmer

To produce safe Java programs you don't need to be a good Java programmer

(To produce a *good* Java program, you do need to be a good programmer)

Why C?

To produce good, safe C programs, you need to be a very good programmer

To produce safe Java programs you don't need to be a good Java programmer

(To produce a *good* Java program, you do need to be a good programmer)

This is an explicit trade-off of speed and flexibility of C against the (relative) safety of managed languages (like Java)

Why C?

To produce good, safe C programs, you need to be a very good programmer

To produce safe Java programs you don't need to be a good Java programmer

(To produce a *good* Java program, you do need to be a good programmer)

This is an explicit trade-off of speed and flexibility of C against the (relative) safety of managed languages (like Java)

It is up to the programmer choosing the language to decide which way they want to go

Why C?

C is a popular example of the *procedural style*

Why C?

C is a popular example of the *procedural style*

It's certainly not the earliest example

Why C?

C is a popular example of the *procedural style*

It's certainly not the earliest example

The procedural style has been around since the 1960s and, until OO came along, was pretty much the only decent way for programmers to structure their programs

Why C?

C is a popular example of the *procedural style*

It's certainly not the earliest example

The procedural style has been around since the 1960s and, until OO came along, was pretty much the only decent way for programmers to structure their programs

For example, FORTRAN (1960s) is procedural

Why C?

You need to see a procedural style language

Why C?

You need to see a procedural style language

You need C for other Units, e.g., Graphics

Why C?

You need to see a procedural style language

You need C for other Units, e.g., Graphics

You should practice writing C programs for yourself; there will also be coding in C in other courses

Why C?

You need to see a procedural style language

You need C for other Units, e.g., Graphics

You should practice writing C programs for yourself; there will also be coding in C in other courses

There is no coursework for C in this Unit: the main practice for coding in C will be (a) self motivated and (b) in other Units

Why C?

Learning C also gives you a big step into learning C++, an object oriented version of C

Why C?

Learning C also gives you a big step into learning C++, an object oriented version of C

Another language that is widely used in industry, it sits somewhere between C and Java

Why C?

Learning C also gives you a big step into learning C++, an object oriented version of C

Another language that is widely used in industry, it sits somewhere between C and Java

In fact, C has been the inspiration or starting point for many languages, e.g., Java, C++, C#, Swift, Rust, and so on, that have tried to fix the “problems” with C

C

Now, C and Java look similar on the page, but are very different languages

Now, C and Java look similar on the page, but are very different languages

C and Java are very different languages

Now, C and Java look similar on the page, but are very different languages

C and Java are very different languages

A lot of what you learned in Java simply does not apply to C

C

The surface syntax is pretty much the same, so we won't spend time on that

C

The surface syntax is pretty much the same, so we won't spend time on that

Such as curly brackets and semicolons, variable declaration and use, function declaration and use; `for`, `while` loops; increment and decrement ++ and `--`; and so on

C

The surface syntax is pretty much the same, so we won't spend time on that

Such as curly brackets and semicolons, variable declaration and use, function declaration and use; `for`, `while` loops; increment and decrement ++ and `--`; and so on

Of course this is because Java adopted C's syntax

C

The surface syntax is pretty much the same, so we won't spend time on that

Such as curly brackets and semicolons, variable declaration and use, function declaration and use; `for`, `while` loops; increment and decrement ++ and `--`; and so on

Of course this is because Java adopted C's syntax

Which C adapted from BCPL

C

The surface syntax is pretty much the same, so we won't spend time on that

Such as curly brackets and semicolons, variable declaration and use, function declaration and use; `for`, `while` loops; increment and decrement ++ and `--`; and so on

Of course this is because Java adopted C's syntax

Which C adapted from BCPL

Which it adapted from Algol, etc.

C

But you need to approach C afresh

C

But you need to approach C afresh

C does not have classes

C

But you need to approach C afresh

C does not have classes

C does not have objects in the OO sense

C

But you need to approach C afresh

C does not have classes

C does not have objects in the OO sense

Warning: I might use the word “object” in a C context. This is the normal English usage of the word meaning “thing”

C

C does not have primitive/non-primitive type distinction

C

C does not have primitive/non-primitive type distinction

This distinction is a Java kludge they needed to get things to work

C

C does not have primitive/non-primitive type distinction

This distinction is a Java kludge they needed to get things to work

In fact, if they had spent a little more time on the language design, they could have eliminated this from Java (see later OO languages for evidence!)

C

C does not have primitive/non-primitive type distinction

This distinction is a Java kludge they needed to get things to work

In fact, if they had spent a little more time on the language design, they could have eliminated this from Java (see later OO languages for evidence!)

In C, all types are primitive!

C

C does not have methods

C does not have methods

C

C has **functions** (aka procedures)

C

C has **functions** (aka procedures)

C has structures (and unions)

C

C has **functions** (aka procedures)

C has structures (and unions)

The principal way of structuring C programs is by the use of functions (thus “procedural style”)

C

C has **functions** (aka procedures)

C has structures (and unions)

The principal way of structuring C programs is by the use of functions (thus “procedural style”)

Don’t be tempted to talk about objects or methods when using C: that will just prove to the listener that you haven’t a clue about C programming

C

C has **functions** (aka procedures)

C has structures (and unions)

The principal way of structuring C programs is by the use of functions (thus “procedural style”)

Don’t be tempted to talk about objects or methods when using C: that will just prove to the listener that you haven’t a clue about C programming

Because of its lack of “sophisticated” constructs, some people call C a low-level language with a high-level language syntax

C

C has **functions** (aka procedures)

C has structures (and unions)

The principal way of structuring C programs is by the use of functions (thus “procedural style”)

Don’t be tempted to talk about objects or methods when using C: that will just prove to the listener that you haven’t a clue about C programming

Because of its lack of “sophisticated” constructs, some people call C a low-level language with a high-level language syntax

That’s not such a bad way of thinking about it

C

C puts as little as possible between the programmer and the machine

C

C puts as little as possible between the programmer and the machine

Unlike Java, which tries to hide the hardware from the programmer: the programmer only sees the abstract “java machine”

C

C puts as little as possible between the programmer and the machine

Unlike Java, which tries to hide the hardware from the programmer: the programmer only sees the abstract “java machine”

Again: this is not a judgement of which language is “better”

C

The only question is “which language is better suited **to the problem in hand**”

C

The only question is “which language is better suited **to the problem in hand**”

This is a common theme of this Unit: no one language is better or worse than another. It’s all a matter of the *right tool for the job*

C

The only question is “which language is better suited **to the problem in hand**”

This is a common theme of this Unit: no one language is better or worse than another. It’s all a matter of the *right tool for the job*

Exercise. Which is the better tool: a screwdriver or a hammer?

C

There is only one way to learn C (or any language)

C

There is only one way to learn C (or any language)

Write C programs

C

There is only one way to learn C (or any language)

Write C programs

Lots of C programs

C

There is only one way to learn C (or any language)

Write C programs

Lots of C programs

If you invest time *now* in writing lots of little example programs, you will reap *hugely* later

C

There is only one way to learn C (or any language)

Write C programs

Lots of C programs

If you invest time *now* in writing lots of little example programs, you will reap *hugely* later

If you don't bother, it will cost you dear later

C

There is only one way to learn C (or any language)

Write C programs

Lots of C programs

If you invest time *now* in writing lots of little example programs, you will reap *hugely* later

If you don't bother, it will cost you dear later

I've lost count of the number of students who have handed in pitiful coursework while saying "I wish I'd had more practice before starting this"

C

Practice, practice, practice

C

Practice, practice, practice

Programming is a very practical skill

C

Practice, practice, practice

Programming is a very practical skill

You didn't learn to drive a car by going to a couple of lectures and looking at a book

C

Practice, practice, practice

Programming is a very practical skill

You didn't learn to drive a car by going to a couple of lectures and looking at a book

You've got to get out and do it

Practice, practice, practice

Programming is a very practical skill

You didn't learn to drive a car by going to a couple of lectures and looking at a book

You've got to get out and do it

Practice!

C

We start with a couple of examples of C programs

```
main(t,_,a )
char
*
a;
{
return!

0<t?
t<3?

main(-79,-13,a+
main(-87,1-_,
main(-86, 0, a+1 )


+a)):

1,
t<_?
main( t+1, _, a )
:3,

main ( -94, -27+t, a )
&&t == 2 ?_
<13 ?

main ( 2, _+1, "%s %d %d\n" )
```

```
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#;#q#
:'d*'3,}{w+K w'K:'+}e#';dq#'l
q#'+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n'){)#}w'){){nl]'/+#n';d}rw'
i;# ){nl]!/n{n#'; r{#w'r nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw'
iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c
;;{nl'-{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ') }+}{rl#'{n'
')# }'+}##(!!/")
:
t<-50?
_==*a ?
putchar(31[a]):

main(-65,_,a+1)
:
main((*a == '/') + t, _, a + 1 )
:

0<t?

main ( 2, 2 , "%s")
:*a=='/'||

main(0,

main(-61,*a, "!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m
.vpbks,fxntdCeghiry")
```

C

Written by Ian Phillipps

C

Written by Ian Phillipps

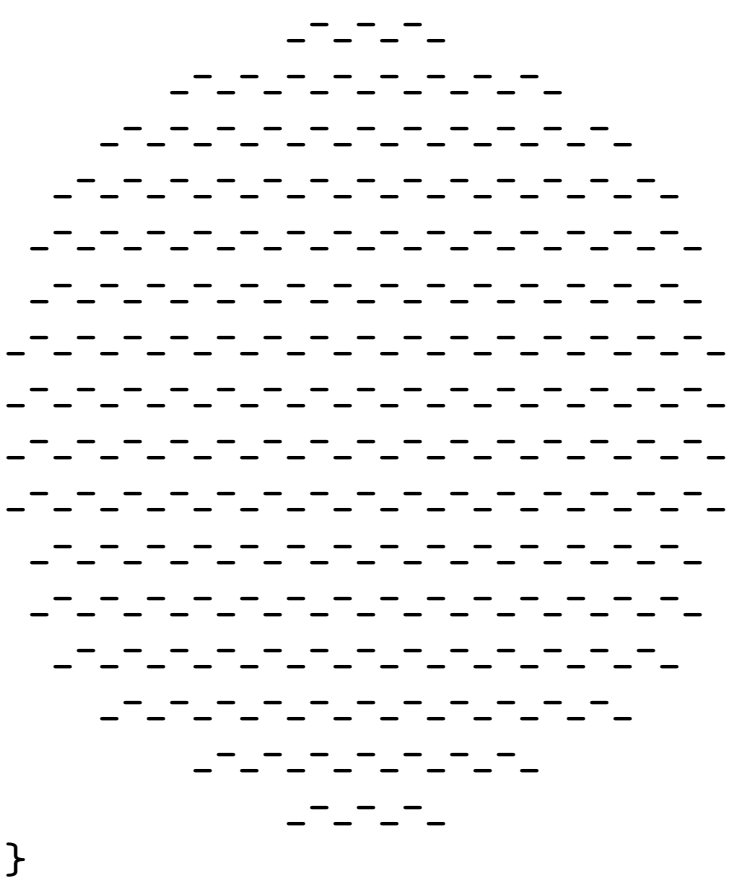
When run this produces...

C

```
On the first day of Christmas my true love gave to me
a partridge in a pear tree.

On the second day of Christmas my true love gave to me
two turtle doves
and a partridge in a pear tree.

On the third day of Christmas my true love gave to me
three french hens, two turtle doves
and a partridge in a pear tree.

On the fourth day of Christmas my true love gave to me
four calling birds, three french hens, two turtle doves
and a partridge in a pear tree.

On the fifth day of Christmas my true love gave to me
five gold rings;
four calling birds, three french hens, two turtle doves
and a partridge in a pear tree.

On the sixth day of Christmas my true love gave to me
six geese a-laying, five gold rings;
four calling birds, three french hens, two turtle doves
...
```

C

```
#define _ F-->00||-F-00--;
int F=00,00=00;main(){F_00();printf("%1.3f\n",4.*-F/00/00);}F_00()
{
```



```
}
```

C

Written by Brian Westley

C

Written by Brian Westley

When run this produces...

C

Written by Brian Westley

When run this produces...

```
3.141
```

C

These are taken from the Obfuscated C Competition: a competition to see how unreadable you can make a C program: `http://www.no.ioccc.org/years.html`

C

These are taken from the Obfuscated C Competition: a competition to see how unreadable you can make a C program: `http://www.no.ioccc.org/years.html`

You can write unreadable programs in all languages: C makes it particularly easy

C

These are taken from the Obfuscated C Competition: a competition to see how unreadable you can make a C program: `http://www.no.ioccc.org/years.html`

You can write unreadable programs in all languages: C makes it particularly easy

So you need to be extra-careful on layout and presentation when writing C!