References & Pointers

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF
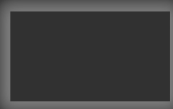
Recap Variables & Memory

- If a variable is declared in the code, a certain amount of memory is allocated for it in the computer's memory

Code level

```
int x = 12;
```

Physical memory level

Address level

`0x7f` Address in memory of variable x

Value level

`12` Value of variable x

Variable name level

`x`

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Reference Type

- A variable of type reference to a data type is an alias of another variable, it refers to an existing variable

Code level

```
int x = 12;

int& ref = x;
```

Physical memory level
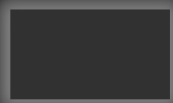
Address level

```
0x7f
```
Address in memory of variable x

Value level

```
12
```
Value of variable x

Variable name level

x, ref

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Reference Type

· A reference variable is an additional name to an existing variable,

· It is not specified in C++ whether actual memory is being allocated for a variable of type reference

Code level

```
int x = 12;

int& ref = x;
```

Physical memory level
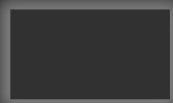
Address level

`0x7f` Address in memory of variable x

Value level

`12` Value of variable x

Variable name level

x, ref

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Reference Type

- This is why a reference variable MUST always be initialized with the object it refers to
- A variable of type reference cannot stand alone

Code level	`int x = 12;`
	`int& ref = x;` `int& refAlone;` ✗

Physical memory level

Address level	`0x7f`	Address in memory of variable x
Value level	`12`	Value of variable x
Variable name level	x, ref	

The Reference Type

- Any change to a variable of type reference will always result in a change of the original variable
- Reference variables can be used just like a normal variable

Code level

```
int x = 12;          ref += 5;
                 => // x (and ref)
int& ref = x;        // will both be 17
```

Physical memory level

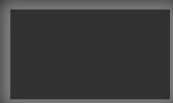Address level

0x7f   Address in memory of variable x

Value level

12   Value of variable x
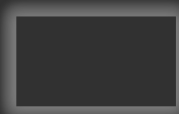
Variable name level

x, ref

The Reference Type by Example

- Manipulating the reference variable directly affects

  the referenced variable & vice versa

```cpp
43  int x = 5;
44  int &z = x;  // z is another name for x
45  int &y;      // Error: reference must be initialized!
46
47  cout << x << endl; // -> prints 5
48  cout << z << endl; // -> prints 5
49
50  z = 9;       // same as x = 9;
51
52  cout << x << endl; // -> prints 9
53  cout << z << endl; // -> prints 9
```

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

Benefits of References

- Reference types support efficient data handling

- In particular, when used as function parameters
  they allow to access data without having
  - to initialize new variables and
  - to copy the variable's values

- Reference types support all kinds of data manipulation whenever
  the referenced variable cannot be accessed directly

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Pointer Type

- A variable of type pointer to a data type is storing

  the address value of an associated variable using &varName

Code level

```
int x = 12;

int* ptr = &x;
```

Physical memory level

Address level	0x7f	Address in memory of variable x	0x2f	Address in memory of variable ptr

Value level	12	Value of variable x	0x7f	Value of variable ptr

Variable name level   x   ptr

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Pointer Type

- The value of the variable the pointer points to can be de-referenced using the dereferencing-operator "*" which must be put in front of the variable's name like so "*ptr"

Accessing the value of the variable ptr points to

Code level

```
int x = 12;          cout << *ptr << endl;
                =>
int* ptr = &x;       // prints 12
```

Physical memory level

Address level	0x7f Address in memory of variable x	0x2f Address in memory of variable ptr
Value level	12 Value of variable x	0x7f Value of variable ptr
Variable name level	x	ptr

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Pointer Type

- The value of the variable the pointer points to can be de-referenced using the dereferencing-operator "*" which must be put in front of the variable's name like so "*ptr"

Changing the variable ptr points to

Code level

```
int x = 12;          *ptr += 5;
              =>      // x (and *ptr)
int* ptr = &x;       // will both be 17
```

Physical memory level

Address level	0x7f	Address in memory of variable x	0x2f	Address in memory of variable ptr
Value level	12	Value of variable x	0x7f	Value of variable ptr
Variable name level	x		ptr	

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

The Pointer Type

- To ensure that pointers to not point into some un-allocated memory block they should always be initialized to a variable or nullptr
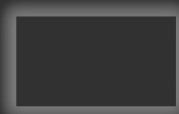
Code level

```
int x = 12;

int* ptr = nullptr;
```

Physical memory level

Address level

0x7f	Address in memory of variable x

0x2f	Address in memory of variable ptr

Value level

12	Value of variable x

0x00	nullptr

Variable name level

x

ptr

Benefits of Pointers

- Pointers in C++ have various application areas

- They are primarily used when the application program

  has to deal with dynamic memory allocation, i.e.,

  - the size of the memory that will be required to store the data

    can only be allocated during run-time

  - the size of the data is not known at compile-time
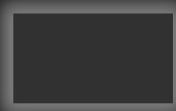
- Such data could be images, audio, user input

References & Pointers Compared

- Similar at first glance, there is a clear difference between the types:
  - References refer to the object they were initialized with
  - Pointers are individual objects that allow to manipulate memory

```cpp
int x = 12;

int& ref = x ;        int* ptr = &x;
```

Address level	0x7f — Address in memory of variable x	0x2f — Address in memory of variable ptr	
Value level	12 — Value of variable x	0x7f — Value of variable ptr	
Variable name level	x, ref	ptr	

FILMUNIVERSITÄT
BABELSBERG
KONRAD WOLF

Prof. Dr.-Ing. Angela Brennecke
Audio & Interactive Media Technologies
Film University Babelsberg KONRAD WOLF

Take Away

- References and pointers are dedicated data types in C++ introduced to increase the efficient management of data

- References should be considered as referring to another object, they are no individual object of themselves

- Pointers are pointing to memory addresses of variables and objects, they are individual objects that support the dynamic and manual allocation and management of data