DIGITAL ARTS & ENTERTAINMENT

Overview course Programming 1

1. Variables 1
2. Variables 2
3. Variables 3
4. Conditionals
5. Iterations
6. Functions 1
7. Functions 2
8. Arrays
9. Strings – Game
10. Classes 1 – Encapsulation
11. Classes 2 – Static const

howest
De Hogeschool West-Vlaanderen
Lid van de Associatie Universiteit Gent

DIGITAL ARTS & ENTERTAINMENT

Variables 3

howest
De Hogeschool West-Vlaanderen
Lid van de Associatie Universiteit Gent

Topics

- representing unsigned numbers
- representing signed numbers
- const variable qualifier
- Scope: local vs global variables

Representing unsigned (positive) numbers

highest significant bit

lowest significant bit

1	0	1	0	1	1	0	1

bit index	7	6	5	4	3	2	1	0
represented value	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1

1 x 128 + 0 x 64 + 1 x 32 + 0 x 16 + 1 x 8 + 1 x 4 + 0 x 2 + 1 x 1

128 + 32 + 8 + 4 + 1 = 173

Range of unsigned primitives

unsigned char: 1 byte → 8 bits → 256 different values

Binary	Decimal	
1111 1101	253	
1111 1110	254	+1
1111 1111	255	+1
?	?	+1

Range of unsigned primitives

unsigned char: 1 byte → 8 bits → 256 different values

		Binary	Decimal	
		1111 1101	253	+1
		1111 1110	254	+1
		1111 1111	255	+1
overflow	1	0000 0000	0	

Range of unsigned primitives

unsigned char: 1 byte → 8 bits → 256 different values

		Binary	Decimal	
		1111 1101	253	+1
		1111 1110	254	+1
		1111 1111	255	+1
overflow	1	0000 0000	0	+1
		0000 0001	1	

demo

- Byte Counter

signed vs unsigned

- For all the integer types there are also unsigned types:

- int → unsigned int
- char → unsigned char

Topics

- representing unsigned numbers
- representing signed numbers
- const variable qualifier
- Scope: local vs global variables

Representing unsigned (positive) numbers

Wikipedia: 2's complement value:

"...to reverse the sign of most integers ... you can take the two's complement of its binary representation. The two's complement is calculated by inverting the digits and adding one. "

-> see sem2 course Game Tech 1

Binary	Decimal
0000 0110	6
1111 1001	invert digits
1111 1010	add one → -6

Two's Complement

- Advantage: arithmetic operations + - / * on signed binary types are identical to those for unsigned binary numbers.
  - CPU design
- Zero has only a single representation, eliminating the subtleties associated with negative zero, which exists in ones'-complement systems.

Range of signed primitives

char: 1byte → 8 bits → 256 different values:

-128 → +127

One sign bit

Binary	Decimal
0111 1110	126
0111 1111	127
1000 0000	-128
1000 0001	-127
1000 0010	-126
1111 1111	-1

Weird problem:

```
char a{ 127 };
a = a + 1;
```

- What is the value of a?

Weird problem:

```
char a{ 127 };
a = a + 1;
```

- What is the value of a? → -128

Weird problem:

char: 1byte → 8 bits → 256 different values:

-128 → +127 → SIGNED

One sign bit → overflow causes sign flip

Binary	Decimal
0111 1110	126
0111 1111	127
1000 0000	-128
1000 0001	-127
1000 0010	-126
1111 1111	-1

demo

- Byte Counter

Topics

- representing unsigned numbers
- representing signed numbers
- const variable qualifier
- Scope: local vs global variables

const variable qualifier

- Variables of which is known their value will never change, should be const.
- C++ Coding standards – rule 15: Use const proactively
- Why? Readability. When others read your code, it makes their life easier when you have const types.
- Examples:

```
const float gravity{9.81f};
const int maxValue{20};
```

Topics

- representing unsigned numbers
- representing signed numbers
- const variable qualifier
- Scope: local vs global variables

Local (stack) variables:

- A local variable is a variable that is declared inside a block: { }
- Always initialize a local variable upon declaration.
- It can only be used in the block in which it was declared.

```
void SomeFunction();
int main()
{
  int number{10};
  ++number;
  SomeFunction();
}
void SomeFunction()
{
  ++number;//error!
}
```

Global variables:

- A global variable is a variable that is declared outside any block: { }
- Always initialize a global variable upon declaration.
- It can be used in any function/block: hence the name "global".
- It has the prefix g_ to easily recognize these global variables.

```
int g_Number{ 10 };
void PrintNumber();
int main()
{
  ++g_Number;
  PrintNumber();
}
void PrintNumber()
{
  std::cout << g_Number;
}
```

Local variables vs global variables

	Local variable	global variable
Lifespan	from declaration till closing }	Entire program
When	preferred	avoid in large projects: makes code unreadable
declaration	in a (member) function	outside functions
initialization	during declaration	during declaration

C++ Coding standards – rule 10: Minimize global and shared data

References

- http://www.cplusplus.com/doc/tutorial/typecasting/
- http://www.cs.cornell.edu/~tomf/notes/cps104/twoscomp.html
- https://en.wikipedia.org/wiki/Two%27s_complement