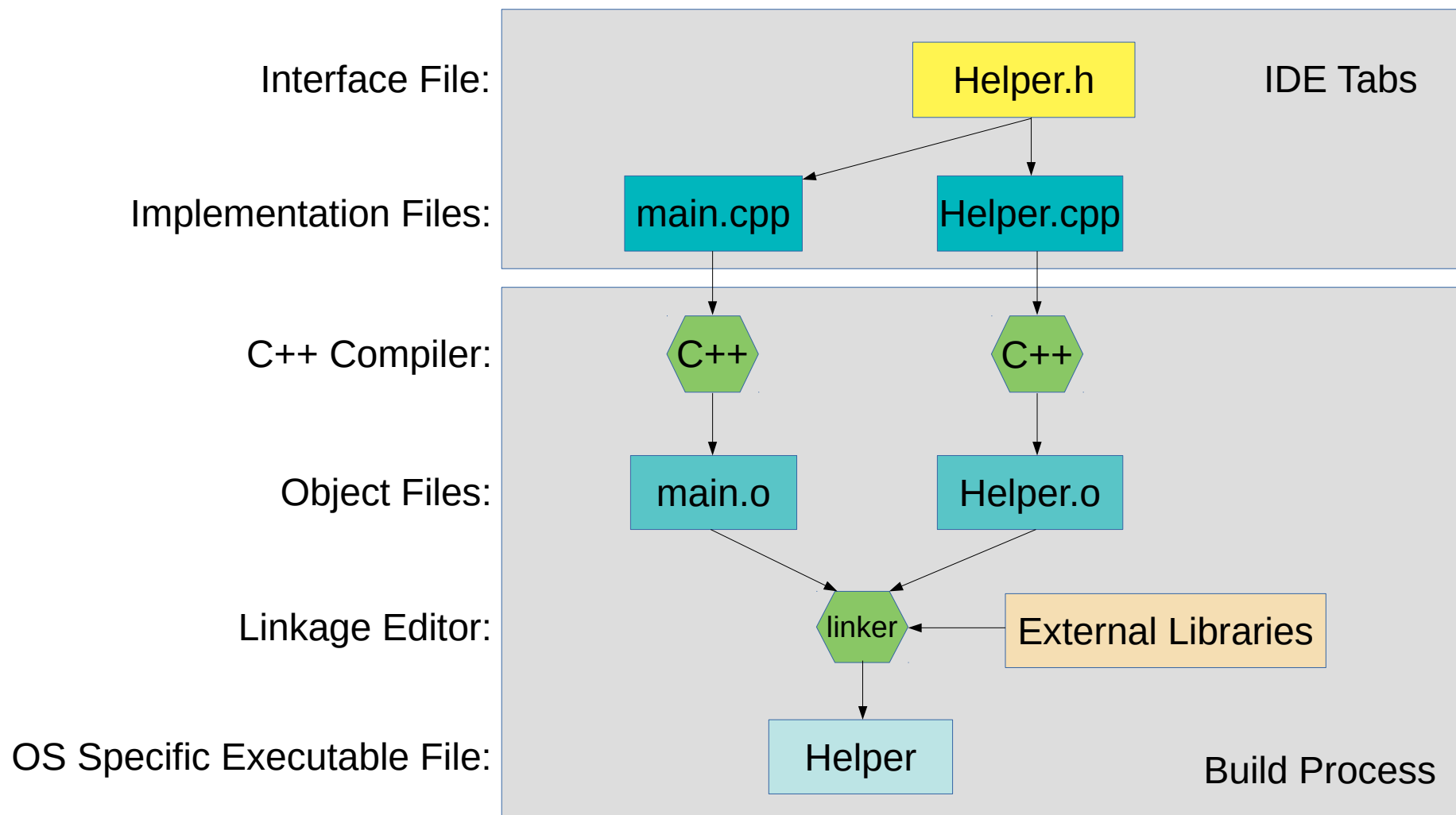


CSCI 390 – Special Topics in C++

Lecture 2

8/23/18

Typical Class Compilation Unit



Getting Help – Provoking Errors

- Read and try to understand the error from the perspective of the compiler/linker.
 - One way of learning how the compiler/linker will respond to an error is to deliberately inject a known error into working code.
- Fix first error and recompile until no more errors.
 - Errors can mask other errors.
 - Errors can confuse the compiler and cause an error storm.
- Compile often to minimize number of errors.
 - This is usually fast and only changed code recompiles.

Provoking Errors

Missing Global

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    const char *Hello = "Hello World";
```

```
    std::cout << DUMPVAR(Hello) << std::endl;
```

```
    return 0;
```

```
}
```

```
// Helper.cpp:
```

```
#include "Helper.h"
```

```
#include <cxxabi.h>
```

```
// Fake unwritten code yet by commenting out code.
```

```
/*
```

```
std::string DemangleName(const char *Name)
```

```
{
```

```
    int Status = -1;
```

```
    std::string DemangledName = Name;
```

```
    char *demangledName = abi::__cxa_demangle (Name, nullptr, nullptr, &Status);
```

```
    if (Status == 0)
```

```
{
```

```
        DemangledName = demangledName;
```

```
}
```

```
    free (demangledName);
```

```
    return DemangledName;
```

```
}
```

```
*/
```

```
// Helper.h:
```

```
#include <cxxabi.h>
```

```
#include <string>
```

```
#include <typeinfo>
```

```
std::string DemangleName(const char *Name);
```

```
#define DUMPTYPE(Symbol) "Type: " << DemangleName(typeid(Symbol).name()) << " , Length: " << sizeof(Symbol)
```

```
#define DUMPVAR(Symbol) "Variable: " << #Symbol << " , " DUMPTYPE(Symbol) << " , Address: " << ((void *) &(Symbol)) << " , Value: " << (Symbol)
```

Provoking Errors

Missing Global - Result

Console:

```
/var/tmp/ccM8td32.o: In function `main':  
main.cpp:(.text.startup+0x26): undefined reference to `DemangleName(char const*)'  
collect2: error: ld returned 1 exit status
```

Provoking Errors

Missing “.”

main.cpp:

```
#include <iostream>

#include "Helper.h"

int main(void)
{
    // Fake missing “.”
    const char *Hello = "Hello World" //;

    std::cout << DUMPVAR(Hello) << std::endl;

    return 0;
}
```

Helper.cpp:

```
#include "Helper.h"
#include <cxxabi.h>

// Fake unwritten code yet by commenting out code.
/*
std::string DemangleName(const char *Name)
{
    int Status = -1;

    std::string DemangledName = Name;
    char *demangledName = abi::__cxa_demangle (Name, nullptr, nullptr, &Status);
    if (Status == 0)
    {
        DemangledName = demangledName;
    }

    free (demangledName);

    return DemangledName;
}
*/
```

Helper.h:

```
#include <cxxabi.h>
#include <string>
#include <typeinfo>

std::string DemangleName(const char *Name);

#define DUMPTYPE(Symbol) "Type: " << DemangleName(typeid(Symbol).name()) << " , Length: " << sizeof(Symbol)

#define DUMPVAR(Symbol) "Variable: " << #Symbol << " , " DUMPTYPE(Symbol) << " , Address: " << ((void *) &(Symbol)) << " , Value: " << (Symbol)
```

Provoking Errors

Missing “;” - Result

Console:

```
main.cpp: In function 'int main()':  
main.cpp:10:5: error: expected ',' or ';' before 'std'  
    std::cout << DUMPVAR(Hello) << std::endl;  
    ^
```

Provoking Errors

Brackets Not Paired

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    const char *Hello = "Hello World";
```

```
    std::cout << DUMPVAR(Hello) << std::endl;
```

```
    return 0;
```

```
}
```

```
// Helper.cpp:
```

```
#include "Helper.h"
```

```
#include <cxxabi.h>
```

```
std::string DemangleName(const char *Name)
```

```
{
```

```
    int Status = -1;
```

```
    std::string DemangledName = Name;
```

```
    char *demangledName = abi::__cxa_demangle (Name, nullptr, nullptr, &Status);
```

```
    if (Status == 0)
```

```
        // Fake missing bracket.
```

```
    // {
```

```
        DemangledName = demangledName;
```

```
    }
```

```
    free (demangledName);
```

```
    return DemangledName;
```

```
}
```

```
// Helper.h:
```

```
#include <cxxabi.h>
```

```
#include <string>
```

```
#include <typeinfo>
```

```
std::string DemangleName(const char *Name);
```

```
#define DUMPTYPE(Symbol) "Type: " << DemangleName(typeid(Symbol).name()) << " , Length: " << sizeof(Symbol)
```

```
#define DUMPVAR(Symbol) "Variable: " << #Symbol << " , " DUMPTYPE(Symbol) << " , Address: " << ((void *) &(Symbol)) << " , Value: " << (Symbol)
```


Provoking Errors

Brackets Not Paired - Result

Console:

Helper.cpp:16:8: error: expected constructor, destructor, or type conversion before '(' token
free (demangledName);
^

Helper.cpp:18:3: error: expected unqualified-id before 'return'
return DemangledName;
^

Helper.cpp:19:1: error: expected declaration before '}' token
}
^

Provoking Errors

Parentheses Not Paired

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    const char *Hello = "Hello World";
```

```
    std::cout << DUMPVAR(Hello) << std::endl;
```

```
    return 0;
```

```
}
```

```
// Helper.cpp:
```

```
#include "Helper.h"
```

```
#include <cxxabi.h>
```

```
std::string DemangleName(const char *Name)
```

```
{
```

```
    int Status = -1;
```

```
    std::string DemangledName = Name;
```

```
    char *demangledName = abi::__cxa_demangle (Name, nullptr, nullptr, &Status);
```

```
    // Open parenthesis missing
```

```
    if Status == 0)
```

```
    {
```

```
        DemangledName = demangledName;
```

```
    }
```

```
    free (demangledName);
```

```
    return DemangledName;
```

```
}
```

```
// Helper.h:
```

```
#include <cxxabi.h>
```

```
#include <string>
```

```
#include <typeinfo>
```

```
std::string DemangleName(const char *Name);
```

```
#define DUMPTYPE(Symbol) "Type: " << DemangleName(typeid(Symbol).name()) << " , Length: " << sizeof(Symbol)
```

```
#define DUMPVAR(Symbol) "Variable: " << #Symbol << " , " DUMPTYPE(Symbol) << " , Address: " << ((void *) &(Symbol)) << " , Value: " << (Symbol)
```

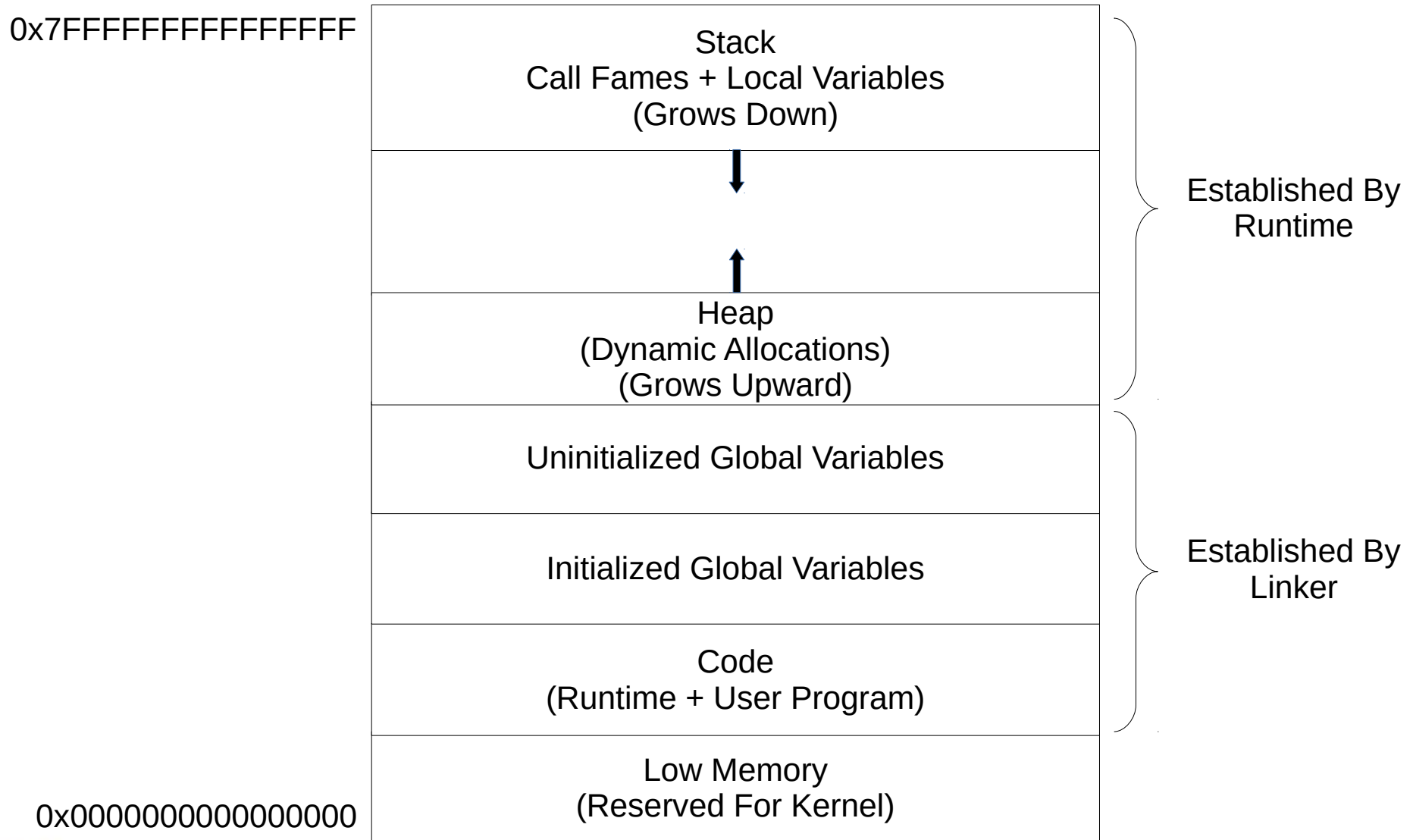
Provoking Errors

Parentheses Not Paired - Result

Console:

```
Helper.cpp: In function 'std::string DemangleName(const char*)':  
Helper.cpp:12:6: error: expected '(' before 'Status'  
    if Status == 0)  
       ^
```

Memory Usage



Intrinsic Types

Overview of Integer Types

Type specifier	Equivalent type
short	short int
short int	
signed short	
signed short int	
unsigned short	unsigned short int
unsigned short int	
int	int
signed	
signed int	
unsigned	unsigned int
unsigned int	
long	long int
long int	
signed long	
signed long int	
unsigned long	unsigned long int
unsigned long int	
long long	long long int (C++11)
long long int	
signed long long	
signed long long int	
unsigned long long	unsigned long long int (C++11)
unsigned long long int	

Intrinsic Types

bool Type

Type	Constants
bool	false, true

// main.cpp:

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    bool b1 = true;
```

```
    bool b2 = false;
```

```
    auto a1 = true;
```

```
    std::cout << DUMPVAR(b1) << std::endl;
```

```
    std::cout << DUMPVAR(b2) << std::endl;
```

```
    std::cout << DUMPVAR(a1) << std::endl;
```

```
    return 0;
```

```
}
```

Console:

Variable: b1, Type: bool, Length: 1, Address: 0x7fff86517bfd, Value: 1

Variable: b2, Type: bool, Length: 1, Address: 0x7fff86517bfe, Value: 0

Variable: a1, Type: bool, Length: 1, Address: 0x7fff86517bfd, Value: 1

...Program finished with exit code 0

Press ENTER to exit console.

Intrinsic Types

char Types

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    signed char c1 = 'a';
```

```
    signed char c2 = '\x61';
```

```
    signed char c3 = '\141';
```

```
    signed char c4 = '\\';
```

```
    char c5 = 'a';
```

```
    char c6 = '\x61';
```

```
    char c7 = '\141';
```

```
    char c8 = '\\';
```

```
    unsigned char c9;
```

```
    auto a1 = 'a';
```

```
    std::cout << DUMPVAR(c1) << std::endl;
```

```
    std::cout << DUMPVAR(c2) << std::endl;
```

```
    std::cout << DUMPVAR(c3) << std::endl;
```

```
    std::cout << DUMPVAR(c4) << std::endl;
```

```
    std::cout << DUMPVAR(c5) << std::endl;
```

```
    std::cout << DUMPVAR(c6) << std::endl;
```

```
    std::cout << DUMPVAR(c7) << std::endl;
```

```
    std::cout << DUMPVAR(c8) << std::endl;
```

```
    std::cout << DUMPVAR(c9) << std::endl;
```

```
    std::cout << DUMPVAR(a1) << std::endl;
```

```
    return 0;
```

```
}
```

Type	Constants
signed char, char	'a', '\x61', '\141', '\\'
unsigned char	none

Escape Sequences

\\ ' \" \? \\ \a \b \f \n \r \t \v

Console:

Variable: c1, Type: signed char, Length: 1, Address: 0x7ffd5a0414d5, Value: a

Variable: c2, Type: signed char, Length: 1, Address: 0x7ffd5a0414d6, Value: a

Variable: c3, Type: signed char, Length: 1, Address: 0x7ffd5a0414d7, Value: a

Variable: c4, Type: signed char, Length: 1, Address: 0x7ffd5a0414d8, Value: '

Variable: c5, Type: char, Length: 1, Address: 0x7ffd5a0414d9, Value: a

Variable: c6, Type: char, Length: 1, Address: 0x7ffd5a0414da, Value: a

Variable: c7, Type: char, Length: 1, Address: 0x7ffd5a0414db, Value: a

Variable: c8, Type: char, Length: 1, Address: 0x7ffd5a0414dc, Value: '

Variable: c9, Type: unsigned char, Length: 1, Address: 0x7ffd5a0414dd, Value:

Variable: a1, Type: char, Length: 1, Address: 0x7ffd5a0414e1, Value: a

...Program finished with exit code 0

Press ENTER to exit console.

Intrinsic Types

short Types

Type	Constants
signed short, short	none
unsigned short	none

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    signed short s1;
```

```
    short s2;
```

```
    unsigned short s3;
```

```
    std::cout << DUMPVAR(s1) << std::endl;
```

```
    std::cout << DUMPVAR(s2) << std::endl;
```

```
    std::cout << DUMPVAR(s3) << std::endl;
```

```
    return 0;
```

```
}
```

Console:

Variable: s1, Type: short, Length: 2, Address: 0x7ffc8ab4eb5a, Value: 0

Variable: s2, Type: short, Length: 2, Address: 0x7ffc8ab4eb5c, Value: 0

Variable: s3, Type: unsigned short, Length: 2, Address: 0x7ffc8ab4eb5e, Value: 0

...Program finished with exit code 0

Press ENTER to exit console.


```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    signed int i1 = 17;  
    signed int i2 = 0x11;  
    signed int i3 = 021;
```

```
    signed i4 = 17;  
    signed i5 = 0x11;  
    signed i6 = 021;
```

```
    int i7 = 17;  
    int i8 = 0x11;  
    int i9 = 021;
```

```
    unsigned int u1 = 17u;  
    unsigned int u2 = 0x11u;  
    unsigned int u3 = 021u;
```

```
    unsigned u4 = 17u;  
    unsigned u5 = 0x11u;  
    unsigned u6 = 021u;
```

```
    auto a1 = 17;  
    auto a2 = 17u;
```

```
    std::cout << DUMPVAR(i1) << std::endl;  
    std::cout << DUMPVAR(i2) << std::endl;  
    std::cout << DUMPVAR(i3) << std::endl;
```

```
    std::cout << DUMPVAR(i4) << std::endl;  
    std::cout << DUMPVAR(i5) << std::endl;  
    std::cout << DUMPVAR(i6) << std::endl;
```

```
    std::cout << DUMPVAR(i7) << std::endl;  
    std::cout << DUMPVAR(i8) << std::endl;  
    std::cout << DUMPVAR(i9) << std::endl;
```

```
    std::cout << DUMPVAR(a1) << std::endl;  
    std::cout << DUMPVAR(a2) << std::endl;
```

```
    return 0;
```

```
}
```

Intrinsic Types

int Types

Type	Constants
signed int, signed, int	17, 0x11, 021
unsigned int, unsigned	17u, 0x11U, 021u

Console:

```
Variable: i2, Type: int, Length: 4, Address: 0x7ffea7617a60, Value: 17  
Variable: i3, Type: int, Length: 4, Address: 0x7ffea7617a64, Value: 17  
Variable: i4, Type: int, Length: 4, Address: 0x7ffea7617a68, Value: 17  
Variable: i5, Type: int, Length: 4, Address: 0x7ffea7617a6c, Value: 17  
Variable: i6, Type: int, Length: 4, Address: 0x7ffea7617a70, Value: 17  
Variable: i7, Type: int, Length: 4, Address: 0x7ffea7617a74, Value: 17  
Variable: i8, Type: int, Length: 4, Address: 0x7ffea7617a78, Value: 17  
Variable: i9, Type: int, Length: 4, Address: 0x7ffea7617a7c, Value: 17  
Variable: u1, Type: unsigned int, Length: 4, Address: 0x7ffea7617a80, Value: 17  
Variable: u2, Type: unsigned int, Length: 4, Address: 0x7ffea7617a84, Value: 17  
Variable: u3, Type: unsigned int, Length: 4, Address: 0x7ffea7617a88, Value: 17  
Variable: u4, Type: unsigned int, Length: 4, Address: 0x7ffea7617a8c, Value: 17  
Variable: u5, Type: unsigned int, Length: 4, Address: 0x7ffea7617a90, Value: 17  
Variable: u6, Type: unsigned int, Length: 4, Address: 0x7ffea7617a94, Value: 17  
Variable: a1, Type: int, Length: 4, Address: 0x7ffea7617a98, Value: 17  
Variable: a2, Type: unsigned int, Length: 4, Address: 0x7ffea7617a9c, Value: 17
```

```
...Program finished with exit code 0  
Press ENTER to exit console.
```

Intrinsic Types

long Types

```
// main.cpp:
#include <iostream>

#include "Helper.h"

int main(void)
{
    signed long i1 = 17L;
    signed long i2 = 0x11L;
    signed long i3 = 021L;

    long i4 = 17L;
    long i5 = 0x11L;
    long i6 = 021L;

    unsigned long u1 = 17ul;
    unsigned long u2 = 0x11UL;
    unsigned long u3 = 021ul;

    auto a1 = 17L;
    auto a2 = 17ul;

    std::cout << DUMPVAR(i1) << std::endl;
    std::cout << DUMPVAR(i2) << std::endl;
    std::cout << DUMPVAR(i3) << std::endl;

    std::cout << DUMPVAR(i4) << std::endl;
    std::cout << DUMPVAR(i5) << std::endl;
    std::cout << DUMPVAR(i6) << std::endl;

    std::cout << DUMPVAR(u1) << std::endl;
    std::cout << DUMPVAR(u2) << std::endl;
    std::cout << DUMPVAR(u3) << std::endl;

    std::cout << DUMPVAR(a1) << std::endl;
    std::cout << DUMPVAR(a2) << std::endl;

    return 0;
}
```

Type	Constants
signed long, long	17L, 0x11L, 021L
unsigned long	17ul, 0x11UL, 021ul

Console:

Variable: i1, Type: long, Length: 8, Address: 0x7ffdfdc40148, Value: 17
Variable: i2, Type: long, Length: 8, Address: 0x7ffdfdc40150, Value: 17
Variable: i3, Type: long, Length: 8, Address: 0x7ffdfdc40158, Value: 17
Variable: i4, Type: long, Length: 8, Address: 0x7ffdfdc40160, Value: 17
Variable: i5, Type: long, Length: 8, Address: 0x7ffdfdc40168, Value: 17
Variable: i6, Type: long, Length: 8, Address: 0x7ffdfdc40170, Value: 17
Variable: u1, Type: unsigned long, Length: 8, Address: 0x7ffdfdc40178, Value: 17
Variable: u2, Type: unsigned long, Length: 8, Address: 0x7ffdfdc40180, Value: 17
Variable: u3, Type: unsigned long, Length: 8, Address: 0x7ffdfdc40188, Value: 17
Variable: a1, Type: long, Length: 8, Address: 0x7ffdfdc40190, Value: 17
Variable: a2, Type: unsigned long, Length: 8, Address: 0x7ffdfdc40198, Value: 17

...Program finished with exit code 0
Press ENTER to exit console.

Intrinsic Types

long long Types

```
// main.cpp:
#include <iostream>

#include "Helper.h"

int main(void)
{
    signed long long i1 = 17LL;
    signed long long i2 = 0x11LL;
    signed long long i3 = 021ll;

    long long i4 = 17LL;
    long long i5 = 0x11LL;
    long long i6 = 021ll;

    unsigned long long u1 = 17ull;
    unsigned long long u2 = 0x11ULL;
    unsigned long long u3 = 021ull;

    auto a1 = 17LL;
    auto a2 = 17ull;

    std::cout << DUMPVAR(i1) << std::endl;
    std::cout << DUMPVAR(i2) << std::endl;
    std::cout << DUMPVAR(i3) << std::endl;

    std::cout << DUMPVAR(i4) << std::endl;
    std::cout << DUMPVAR(i5) << std::endl;
    std::cout << DUMPVAR(i6) << std::endl;

    std::cout << DUMPVAR(u1) << std::endl;
    std::cout << DUMPVAR(u2) << std::endl;
    std::cout << DUMPVAR(u3) << std::endl;

    std::cout << DUMPVAR(a1) << std::endl;
    std::cout << DUMPVAR(a2) << std::endl;

    return 0;
}
```

Type	Constants
signed long long, long long	17LL, 0x11LL, 021ll
unsigned long	17ull, 0x11ULL, 021ull

Console:

Variable: i1, Type: long long, Length: 8, Address: 0x7ffe0dec2688, Value: 17
Variable: i2, Type: long long, Length: 8, Address: 0x7ffe0dec2690, Value: 17
Variable: i3, Type: long long, Length: 8, Address: 0x7ffe0dec2698, Value: 17
Variable: i4, Type: long long, Length: 8, Address: 0x7ffe0dec26a0, Value: 17
Variable: i5, Type: long long, Length: 8, Address: 0x7ffe0dec26a8, Value: 17
Variable: i6, Type: long long, Length: 8, Address: 0x7ffe0dec26b0, Value: 17
Variable: u1, Type: unsigned long long, Length: 8, Address: 0x7ffe0dec26b8, Value: 17
Variable: u2, Type: unsigned long long, Length: 8, Address: 0x7ffe0dec26c0, Value: 17
Variable: u3, Type: unsigned long long, Length: 8, Address: 0x7ffe0dec26c8, Value: 17
Variable: a1, Type: long long, Length: 8, Address: 0x7ffe0dec26d0, Value: 17
Variable: a2, Type: unsigned long long, Length: 8, Address: 0x7ffe0dec26d8, Value: 17

...Program finished with exit code 0
Press ENTER to exit console.

Intrinsic Types

Floating Point Types

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    float f1 = 1.0f;
```

```
    double f2 = 2.0;
```

```
    auto a1 = 1.0f;
```

```
    auto a2 = 2.0;
```

```
    std::cout << DUMPVAR(f1) << std::endl;
```

```
    std::cout << DUMPVAR(f2) << std::endl;
```

```
    std::cout << DUMPVAR(a1) << std::endl;
```

```
    std::cout << DUMPVAR(a2) << std::endl;
```

```
    return 0;
```

```
}
```

Type	Constants
float	1.0f
double	1.0
long double	1.0L

Console:

Variable: f1, Type: float, Length: 4, Address: 0x7ffc18eeb498, Value: 1

Variable: f2, Type: double, Length: 8, Address: 0x7ffc18eeb4b0, Value: 2

Variable: a1, Type: float, Length: 4, Address: 0x7ffc18eeb49c, Value: 1

Variable: a2, Type: double, Length: 8, Address: 0x7ffc18eeb4b8, Value: 2

...Program finished with exit code 0

Press ENTER to exit console.

Intrinsic Types

enum Types

```
// main.cpp:
```

```
#include <iostream>
```

```
#include "Helper.h"
```

```
int main(void)
```

```
{
```

```
    enum eYesNo
```

```
    {
```

```
        No,
```

```
        Yes,
```

```
    };
```

```
    eYesNo IsCSCI390 = Yes;
```

```
    eYesNo IsOnlineCourse = No;
```

```
    auto TaughByDavid = Yes;
```

```
    std::cout << DUMPVAR(IsCSCI390) << std::endl;
```

```
    std::cout << DUMPVAR(IsOnlineCourse) << std::endl;
```

```
    std::cout << DUMPVAR(TaughByDavid) << std::endl;
```

```
    return 0;
```

```
}
```

Console:

Variable: IsCSCI390, Type: main::eYesNo, Length: 4, Address: 0x7ffc72f65234, Value: 1

Variable: IsOnlineCourse, Type: main::eYesNo, Length: 4, Address: 0x7ffc72f65238, Value: 0

Variable: TaughByDavid, Type: main::eYesNo, Length: 4, Address: 0x7ffc72f6523c, Value: 1

...Program finished with exit code 0

Press ENTER to exit console.