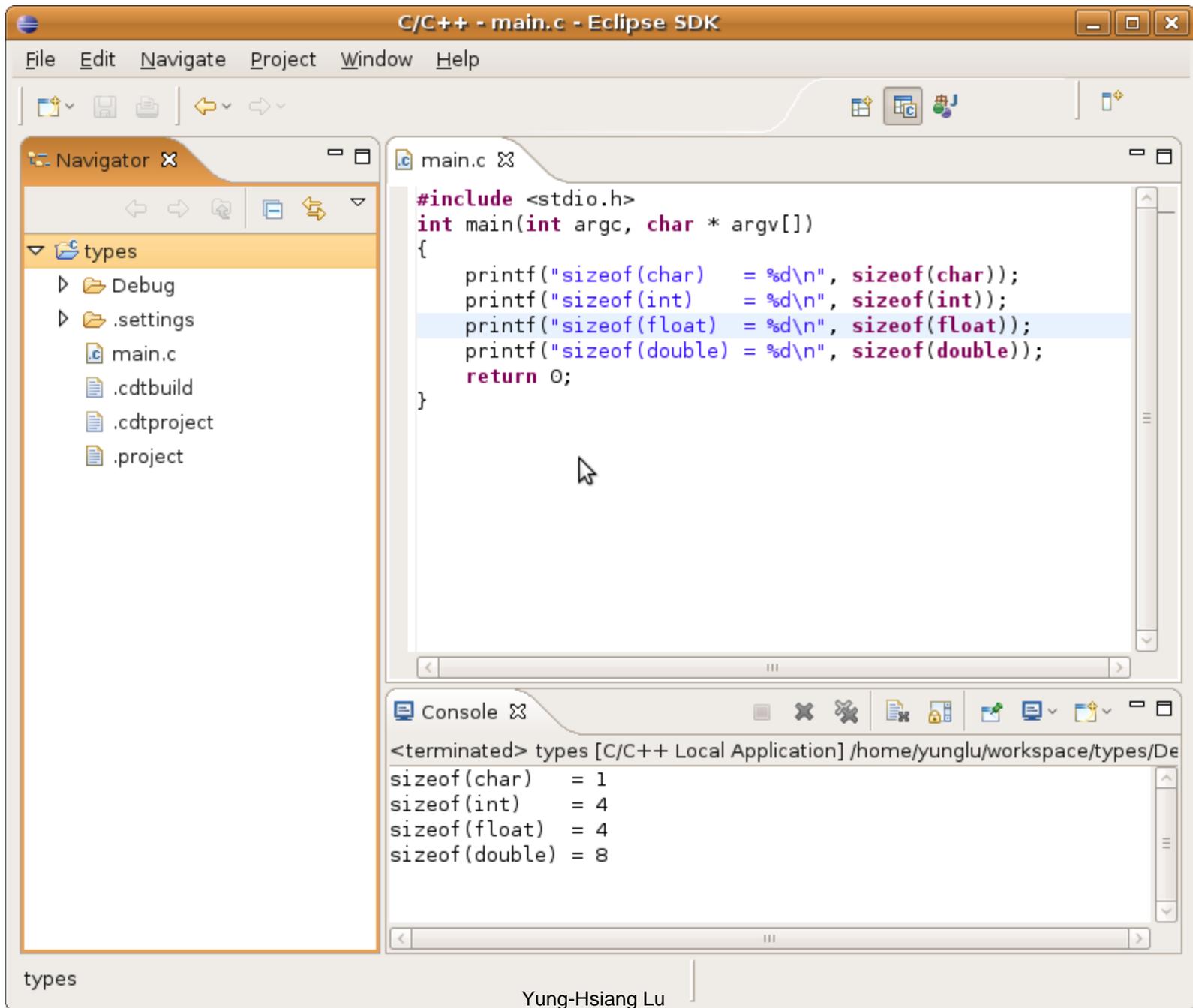


Data Types

Yung-Hsiang Lu

Why Data Types?

- Data types tell C how to handle data and prevent meaningless operations.
- Examples:
 - int
 - char
 - float
 - double
 - unsigned
- Different types may have different sizes (numbers of bytes) and different ranges of values



conversion is based on the ASCII (American Standard Code for Information Interchange) table

: = 58, ? = 63, @ = 64

A = 65, B = 66, C = 67 ...

a = 97, b = 98, c = 99 ...

The screenshot shows an IDE window with a project named 'types'. The left sidebar lists files: 'main.c', '.cdtbuild', '.cdtproject', and '.project'. The main editor displays the following C++ code:

```
printf("sizeof(char) = %d\n", sizeof(char));  
char cv1 = 'C';  
char cv2 = cv1 + 2;  
printf("%c %c\n", cv1, cv2);  
return 0;  
}
```

The 'Console' window at the bottom shows the output of the program:

```
<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/De  
sizeof(char) = 1  
sizeof(int) = 4  
sizeof(float) = 4  
sizeof(double) = 8  
C E ←
```

A yellow arrow points to the output 'C E'. The IDE title bar at the bottom reads 'types' and 'Yung-Hsiang Lu'.

'char' has only one byte, storing -128 to 127.

$65 + 128 = 193 > 127 \Rightarrow$ **overflow**

'int' has four bytes, storing -2,147,483,648 to 2,147,483,647

$65 + 128 = 193 < 2,147,483,647 \Rightarrow$ no problem

The screenshot shows a C++ IDE with a project named 'types'. The source code in 'main.c' is as follows:

```
printf("sizeof(double) = %d\n", sizeof(double));  
char cv1 = 'C';  
char cv2 = cv1 + 2;  
printf("%c %c\n", cv1, cv2);  
char cv3 = 'A' + 128;  
printf("'A' %d + 128 = %d\n", 'A', cv3);  
int iv1 = 'A' + 128;  
printf("'A' %d + 128 = %d\n", 'A', iv1);  
return 0;  
}
```

The console output shows the following results:

```
<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/De  
sizeof(float) = 4  
sizeof(double) = 8  
C E  
'A' 65 + 128 = -63  
'A' 65 + 128 = 193
```

A blue arrow points to the output line for the 'int' calculation, which shows '193', indicating that no overflow occurred for the 'int' type.

types/Debug/types | Yung-Hsiang Lu

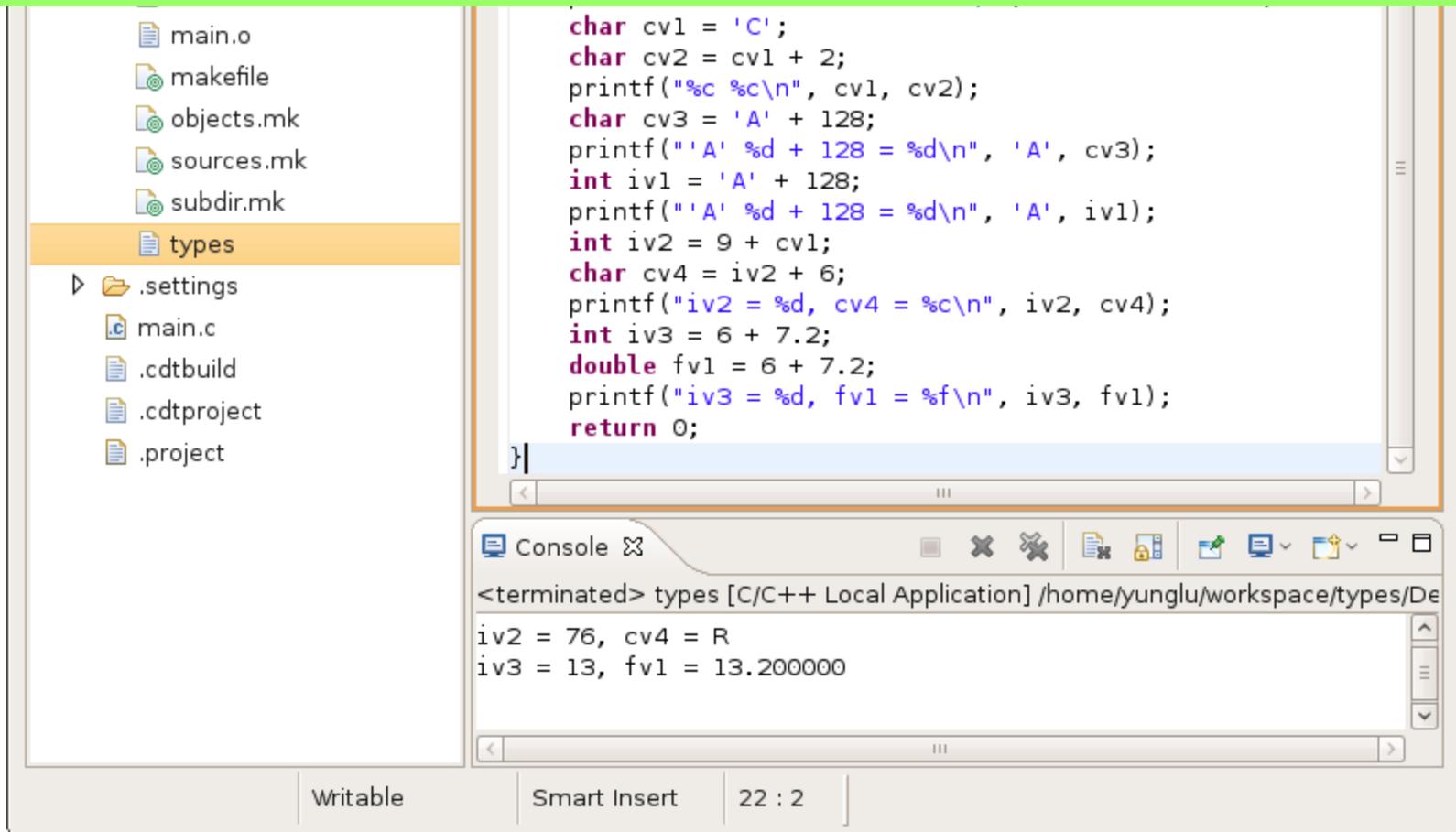
overflow (2's complement)

How can 193 become -63?

	128	64	32	16	8	4	2	1
193 = 128 + 64 + 1 =	0	1	1	0	0	0	0	1
flip the bits	1	0	0	1	1	1	1	0
add one	1	0	0	1	1	1	1	1
ignore the leading one	1	0	0	1	1	1	1	1
compute the value	$63 = 32 + 16 + 8 + 4 + 2 + 1$							
add negative sign	- 63							

assigning a floating point number (7.2) to an integer (iv3) loses the fraction.

To keep the fraction, use double (or float).



The screenshot shows the Eclipse IDE interface. On the left is the Project Explorer showing a project named 'types' with files like main.c, .cdtbuild, .cdtproject, and .project. The main editor window displays the following C++ code:

```
char cv1 = 'C';
char cv2 = cv1 + 2;
printf("%c %c\n", cv1, cv2);
char cv3 = 'A' + 128;
printf("'A' %d + 128 = %d\n", 'A', cv3);
int iv1 = 'A' + 128;
printf("'A' %d + 128 = %d\n", 'A', iv1);
int iv2 = 9 + cv1;
char cv4 = iv2 + 6;
printf("iv2 = %d, cv4 = %c\n", iv2, cv4);
int iv3 = 6 + 7.2;
double fv1 = 6 + 7.2;
printf("iv3 = %d, fv1 = %f\n", iv3, fv1);
return 0;
}
```

Below the code editor is the Console window, which shows the output of the program:

```
<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/De
iv2 = 76, cv4 = R
iv3 = 13, fv1 = 13.200000
```

The status bar at the bottom indicates 'Writable', 'Smart Insert', and '22 : 2'.

$$(a + b) - a \neq b$$

$$dv2 = 1.9 \quad 10^{300}, \quad dv3 = -2.4 \quad 10^{-100}$$

$$dv4 = (dv2 + dv3) - dv2 \text{ and } dv4 = 0.0$$

The screenshot shows the Eclipse IDE with a C++ project named 'types'. The source file 'main.c' is open, displaying the following code:

```

main() {
    printf("iv2 = %d, cv4 = %c\n", iv2, cv4);
    int iv3 = 6 + 7.2;
    double dv1 = 6 + 7.2;
    printf("iv3 = %d, dv1 = %f\n", iv3, dv1);
    double dv2 = 1.9e+300;
    double dv3 = -2.4e-100;
    double dv4 = (dv2 + dv3) - dv2;
    printf("dv2 = %e, dv3 = %e, dv4 = %e\n", dv2, dv3, dv4);
    printf("dv4 - dv3 = %e\n", dv4 - dv3);
    return 0;
}

```

The console output shows the results of the program execution:

```

<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/Debug/types (08/1
dv2 = 1.900000e+300, dv3 = -2.400000e-100, dv4 = 0.000000e+00
dv4 - dv3 = 2.400000e-100

```

The status bar at the bottom indicates 'Writable', 'Smart Insert', and the time '17:46'.

<< means shift left

$$1 = 2^0, 1 \ll 1 \Rightarrow 10 \text{ (binary)} = 2^1$$

$$1 \ll 31 \Rightarrow 10 \dots 0 \text{ (31 zeros, binary)} = 2^{31}$$

int has 4 bytes (32 bits), range -2^{31} to $2^{31} - 1$

$2^{31} + 1$ is positive if considered as unsigned
if it is signed, it becomes negative

```

double dv2 = 1.9e+300;
double dv3 = -2.4e-100;
double dv4 = (dv2 + dv3) - dv2;
printf("dv2 = %e, dv3 = %e, dv4 = %e\n", dv2, dv3, dv4);
printf("dv4 - dv3 = %e\n", dv4 - dv3);
int iv4 = (1 << 31) + 1;
printf("iv4 u = %u, d = %d\n", iv4, iv4);
return 0;
}

```

Console output:

```

<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/Debug/types (08/1
dv4 - dv3 = 2.4000000e-100
iv4 u = 2147483649, d = -2147483647

```

What is correct about char and int types?

- A) They have the same numbers of bytes, both using 4 bytes.
- B) Conversion between char and int is not allowed.
- C) Integer (int) can hold only positive values.
- D) Character (char) can hold only positive values.
- E) None of the above.

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

What is the binary representation of 69 using 8 bits?

- A) 1010 1011
- B) 0100 0101
- C) 0111 0010
- D) 0001 1001
- E) 1100 1101

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

What is the 2's complement of 69 using 8 bits?

- A) 1011 1011
- B) 0101 0010
- C) 1010 0101
- D) 1111 0011
- E) 1001 0101

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

Left Shift

What is the value of $(5 \ll 2) + 9$?

Correct - Click anywhere to
continue

Incorrect - Click anywhere to
continue

Your answer:

You did not answer this question

You must answer the question
before continuing

Submit

Clear

Data Types

Your Score	{score}
Max Score	{max-score}
Number of Quiz Attempts	{total-attempts}

Question Feedback/Review Information Will Appear Here

Continue

Review Quiz