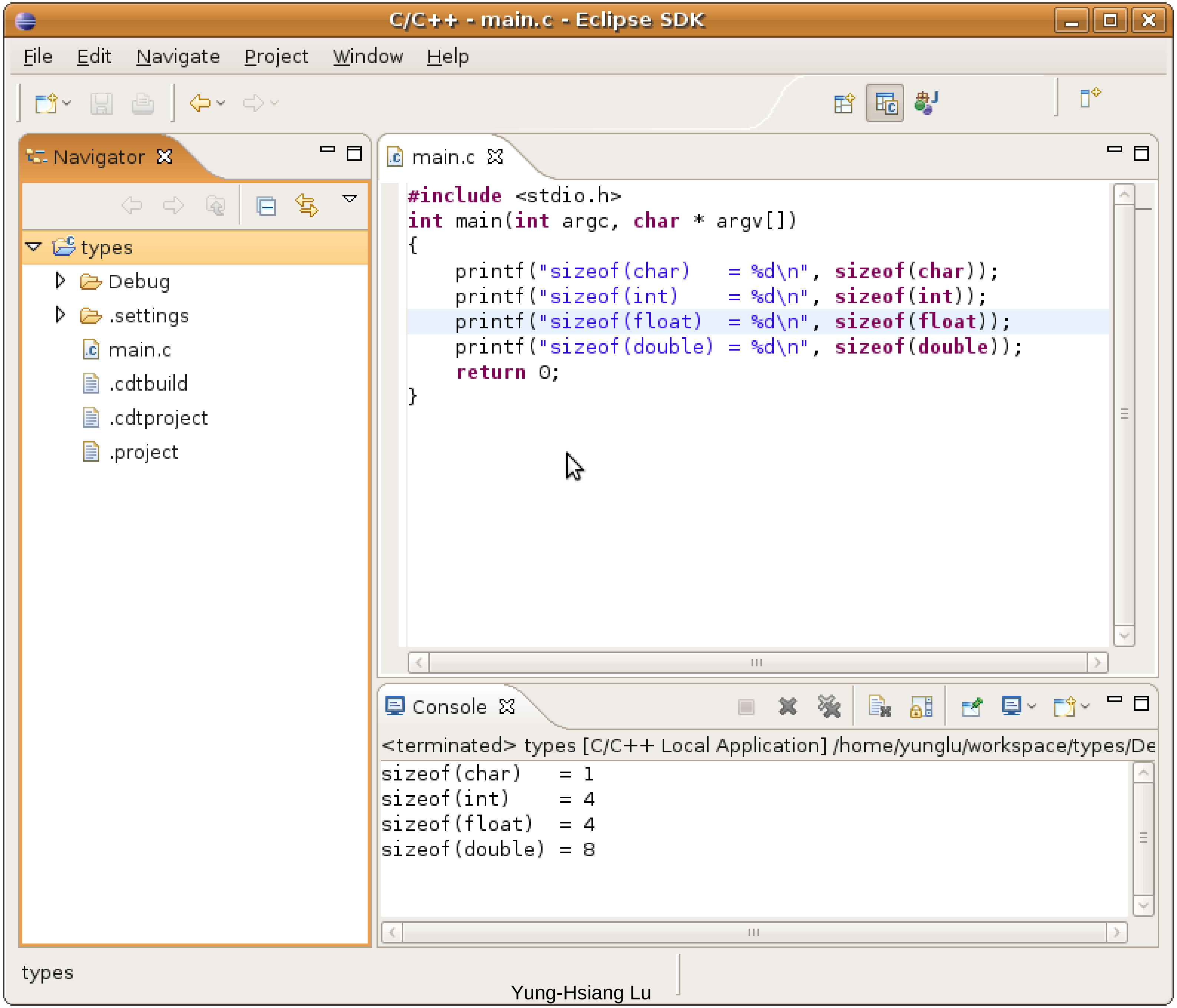
Data Types

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



```
#include <stdio.h>
int main(int argc, char * argv[])
{
    printf("sizeof(char)   = %d\n", sizeof(char));
    printf("sizeof(int)    = %d\n", sizeof(int));
    printf("sizeof(float)  = %d\n", sizeof(float));
    printf("sizeof(double) = %d\n", sizeof(double));
    return 0;
}
```

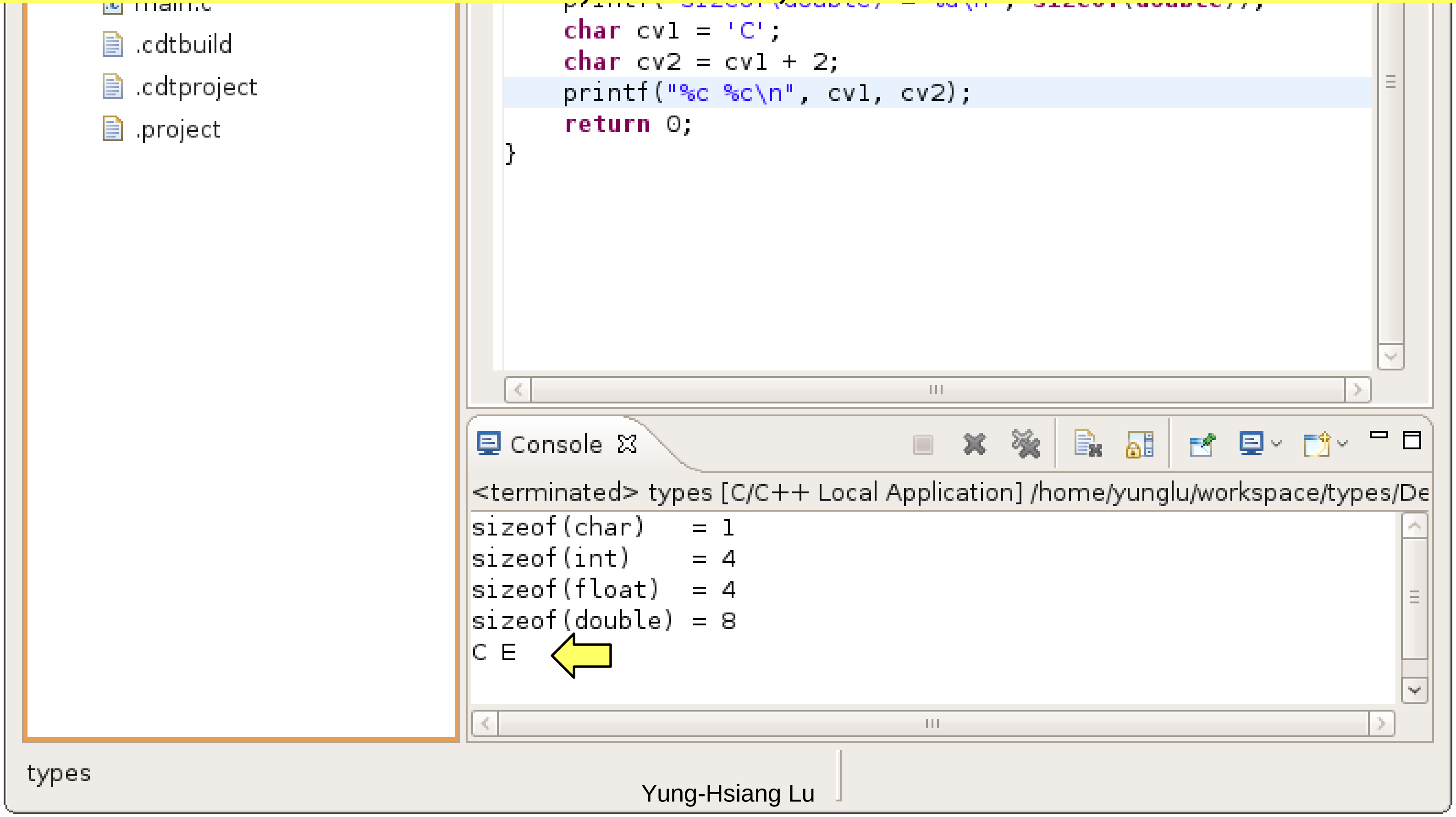
```
<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/De
sizeof(char)   = 1
sizeof(int)    = 4
sizeof(float)  = 4
sizeof(double) = 8
```

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 …



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

<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/De

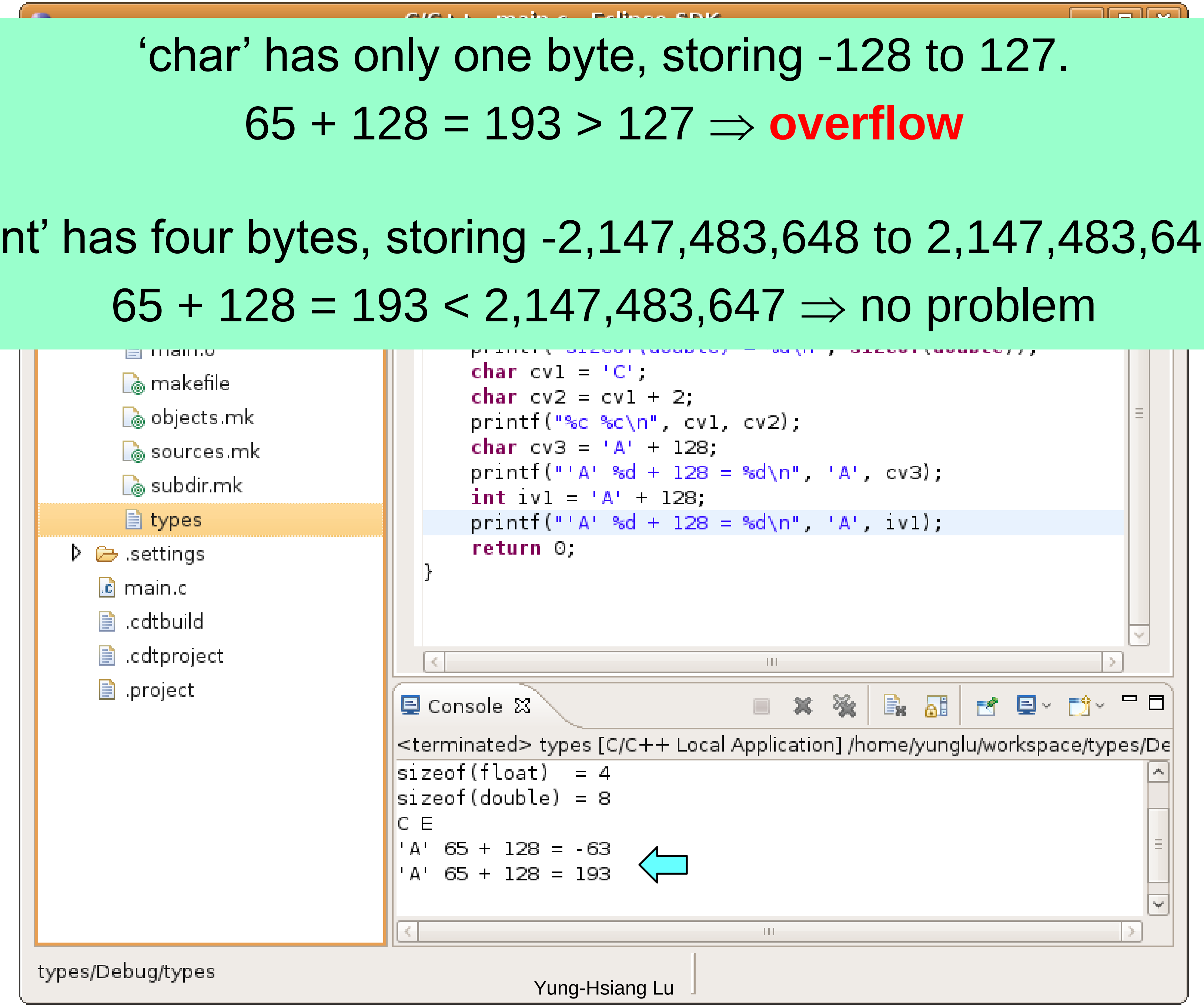
```
sizeof(char)   = 1
sizeof(int)    = 4
sizeof(float)  = 4
sizeof(double) = 8
C E
```

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

65 + 128 = 193 > 127 ⇒ **overflow**

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

65 + 128 = 193 < 2,147,483,647 ⇒ no problem



```
    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;
}
```

<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
```

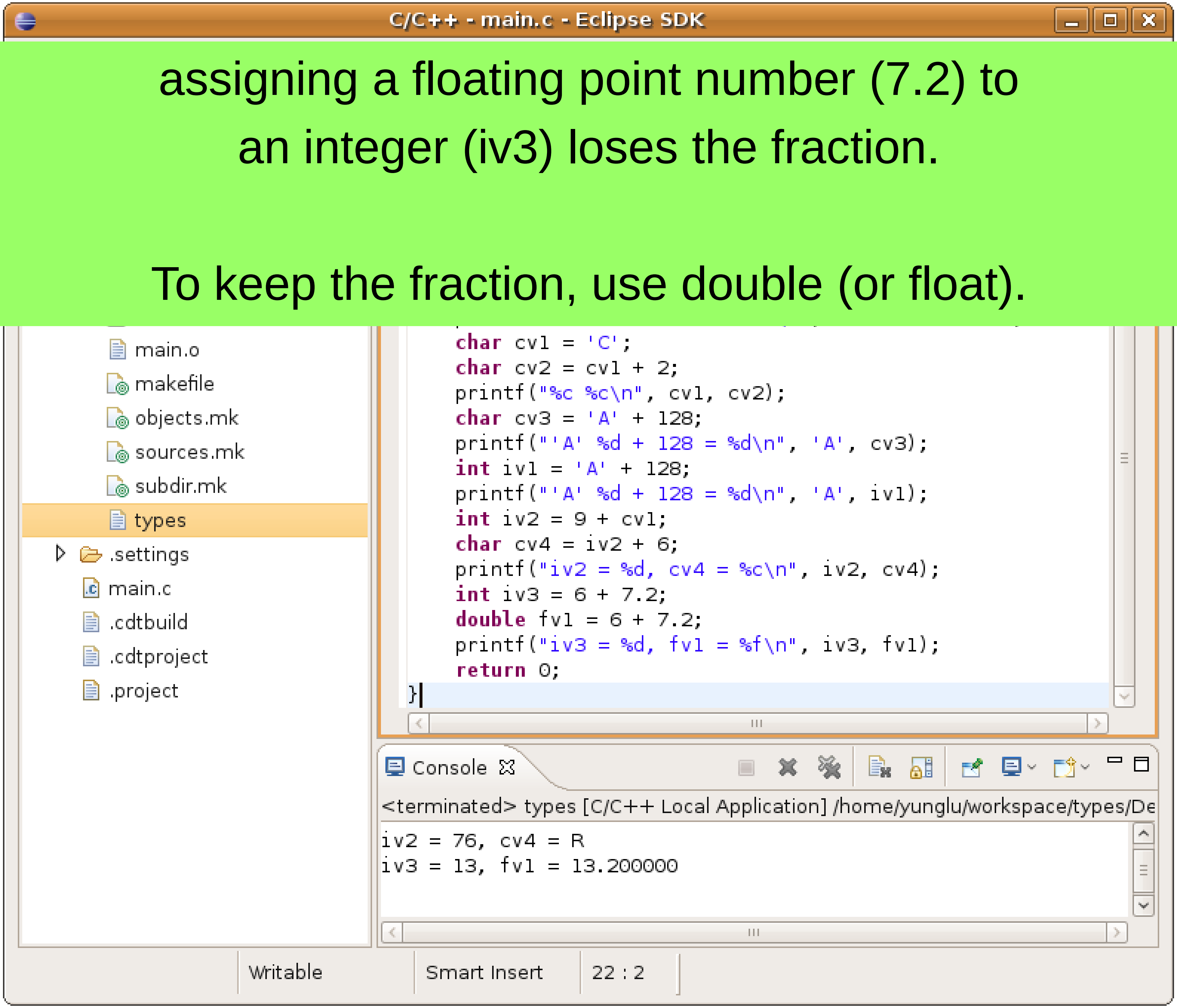
overflow (2's complement)

How can 193 become -63?

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

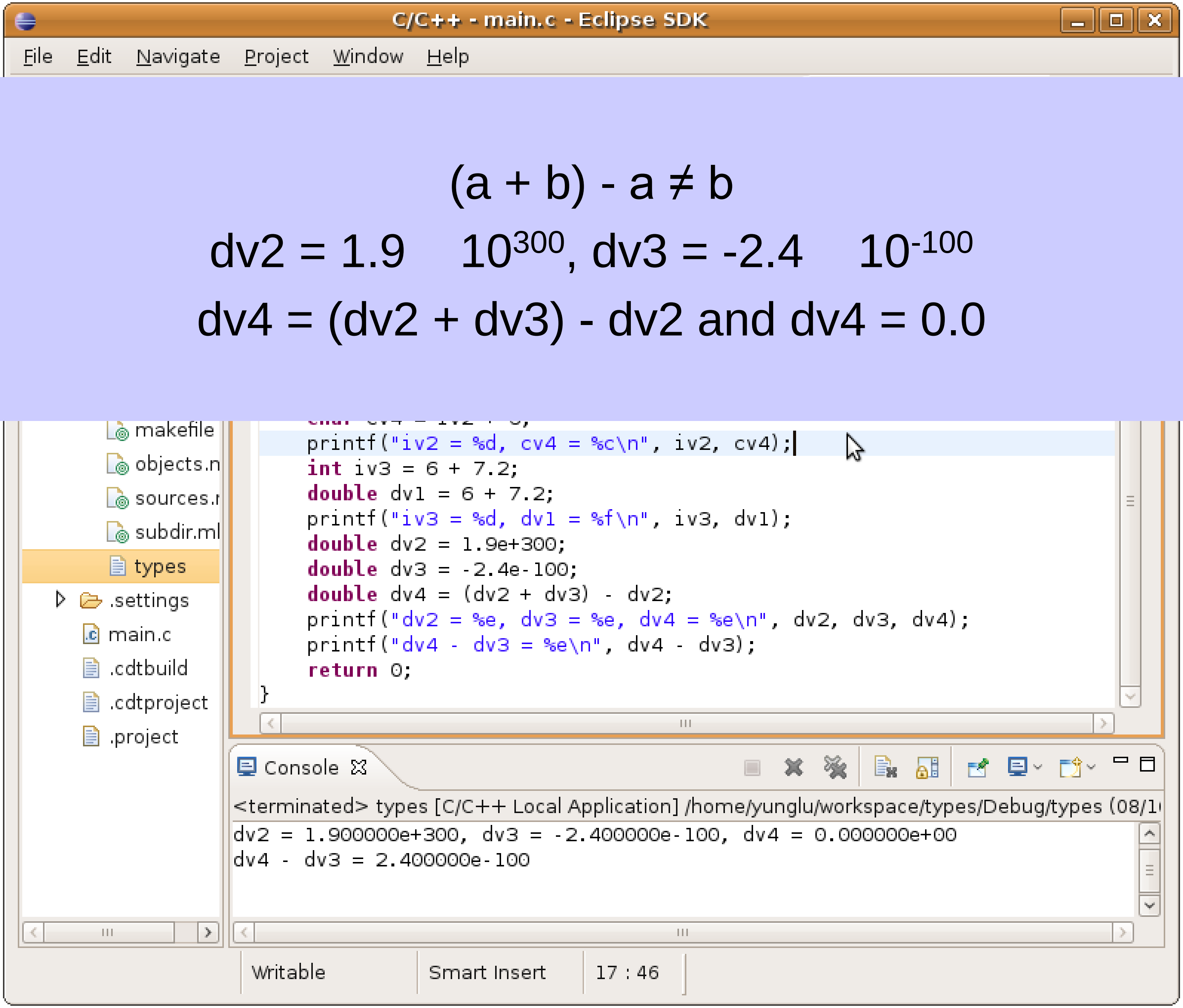
assigning a floating point number (7.2) to an integer (iv3) loses the fraction.

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



```
    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;
}
```

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



(a + b) - a ≠ b

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

dv4 = (dv2 + dv3) - dv2 and dv4 = 0.0

```
    char cv4 = iv2 + 5;
    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;
}
```

```
<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
```

<< means shift left

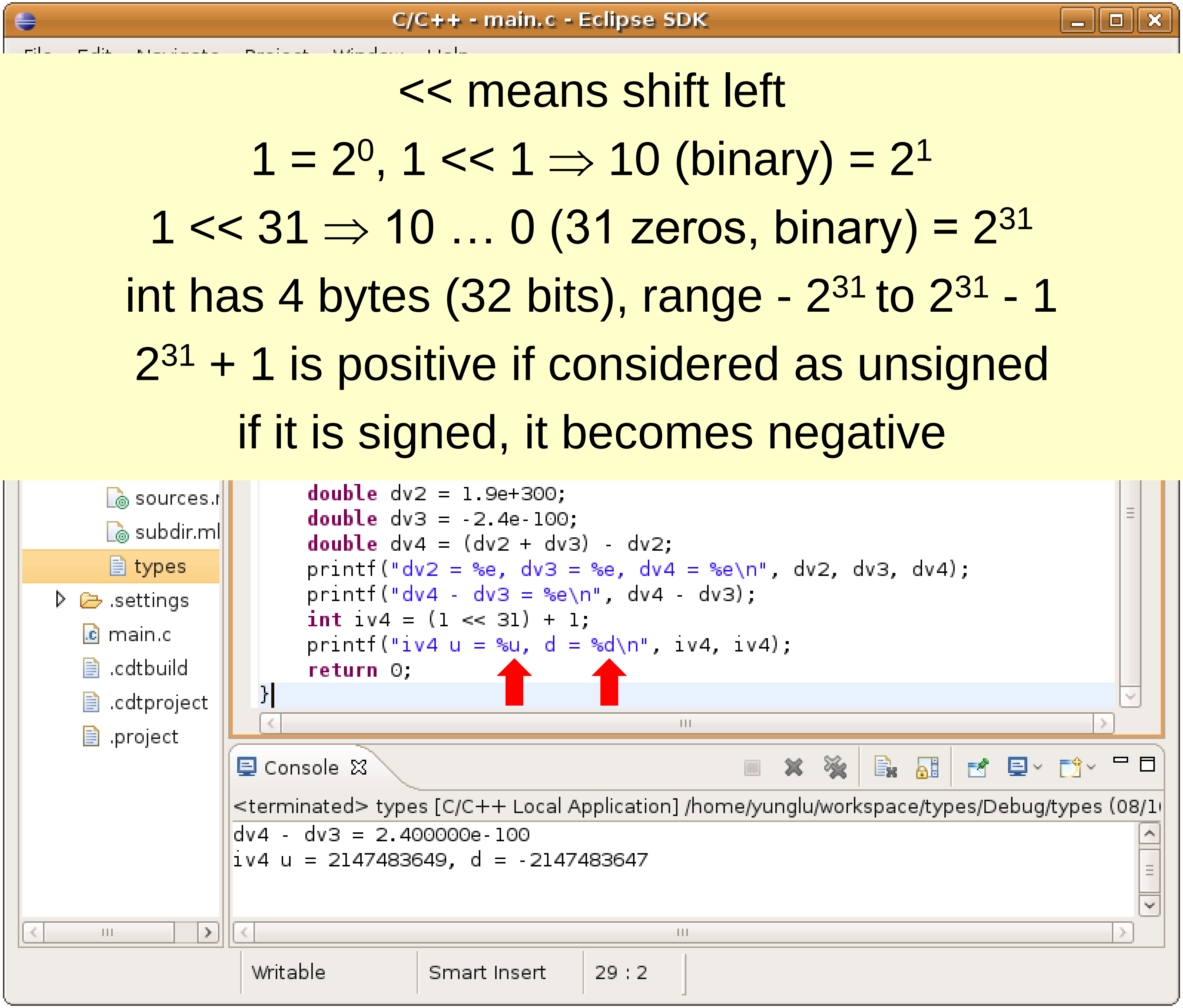
1 = 2^0, 1 << 1 ⇒ 10 (binary) = 2^1

1 << 31 ⇒ 10 … 0 (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;
}
```

```
<terminated> types [C/C++ Local Application] /home/yunglu/workspace/types/Debug/types (08/1
dv4 - dv3 = 2.400000e-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 conti

Incorrect - Click anywhere to

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 conti

Incorrect - Click anywhere to

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 cont

Incorrect - Click anywhere to

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 << 2) + 9?

Correct - Click anywhere to
conti

Incorrect - Click anywhere to

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