Base 2

- Base: 0, 1
- Add the next significant digit when adding 1 to a ‘1’
- Most basic calculation unit of computing science

Base 8

- Base: 0, 1, 2, 3, 4, 5, 6, 7
- Add the next significant digit when adding 1 to a ‘7’
- Used in old computer systems

Base 10

- Base: 0, 1, 2, 3, 4, 5, 6, 7, 8 , 9
- Add the next significant digit when adding 1 to a ‘9’
- Foundation of maths

Base 16

- Base: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Add the next significant digit when adding 1 to a ‘F’
- Foundation of modern 64 bit and 32 bit OS systems

Convert Every other base representation to base 10

```
Value * base ^ (highest significantIndex)
+NextValue * base ^ (highest significantIndex - 1)
+….
+LastValue * base ^ (0)
```

1234_{10} to $????_{10}$

1 * (10 ^ 3) + 2 * (10 ^ 2) + 3 * (10 ^ 1) + 4 * (10 ^ 0) = 1234_{10}

1111_{2} to $????_{10}$

1 * 2 ^ 3 + 1 * 2 ^ 2 + 1 * 2 ^ 1 + 1 * 2 ^ 0 = 8_{10} + 4_{10} + 2_{10} + 1_{10} = 15_{10}

1234_{8} to $????_{10}$

1 * (8 ^ 3) + 2 * (8 ^ 2) + 3 * (8 ^ 1) + 4 * (8 ^ 0) = 668_{8}

1234_{16} to $????_{10}$

1 * (16 ^ 3) + 2 * (16 ^ 2) + 3 * (16 ^ 1) + 4 * (16 ^ 0) = 4660_{16}

$ABCD_{16}$ to $????_{10}$

A * (16 ^ 3) + B * (16 ^ 2) + C * (16 ^ 1) + D * (16 ^ 0) =
10_{10} * (16 ^ 3) + 11_{10} * (16 ^ 2) + 12_{10} * (16 ^ 1) + 13_{10} * (16 ^ 0) = 43981

Convert base 10 to every other base

- Step 1: Take the other base as divider
- Step 2: Use Decimal value mod divider
- Step 3: Write down mod value
- If the remaining decimal value is still larger than divider
  - Repeat step 2
  - otherwise write down last mod value

2 35 1

2 17 1

2 8 0

2 4 0

2 2 0

2 1 1

100011_2

8 35 3

8 4 4

43_8

16 35 3

16 2 2

23_{16}

Convert base 8 to binary

- The highest single digit of base 8 is 7
- 7 can be represent with just 3 bits: 111
- To convert base 8 to binary, calculate each digit to a 3 bits binary and combine them.
- e.g. 45_8, 4 convert to 100_2 , 5 to 101_2 result is 100101_2

Convert base 16 to binary

- The highest single digit of base 8 is F
- F maps to 15 in base 10,
- F can be represent with just 4 bits: 1111
- To convert base 16 to binary, calculate each digit to a 4 bits binary and combine them.
- e.g. AB_8, A convert to 1010_2 , B to 1011_2 result is 10101011_2

Convert base 16 to base 8

- Convert base 16 to binary, calculate each digit to a 4 bits binary and combine them.

- Regroup the binary to a new group of 3 bits, make up the missing digits with 0

- e.g. AB_8, A convert to 1010_2, B to 1011_2 result is 10101011_2, Regroup $010/101/011_2$ result is 253_8

Convert base 8 to base 16

- Convert base 8 to binary, calculate each digit to a 3 bits binary and combine them.

- Regroup the binary to a new group of 4 bits, make up the missing digits with 0

- e.g. 253_8, result is 010101011_2, Regroup 0/1010/1011_2 result is AB_8

Java Primitive Assignment

- You can always assign value from a lower storage cost to a higher storage cost variable
- Floating points always larger than non floating points
    - You can assign any non floating points value to any floating points variable
    - No other way around

Java Primitive Cast

- Cast the the operation to convert the target data type to the assigned data type
- <primitive type> var = (primitive type) targetVar
- e.g.
    - short sVar = 19;
    - int var = (int) sVar;

- Cast cross all the boundary. For safety use
  - Use cast follow by the assignment rule
- Cast floating points (safely) to non floating points will lose all the digit
- Cast non floating points (safely) to floating points, will add .0

```
double d = 888.0d;
long longValue = (long) d;
//longValue will be printed out with 888
```

```
int intValue = 223;
double d = (double) intValue;
//d will be printed out with 223.0
```

```
int intValue = 5;
double d =  (double) intValue  /  2;                2.5
```

```
int intValue = 5;                                    2.0
double d =  (double) (intValue  /  2);
```