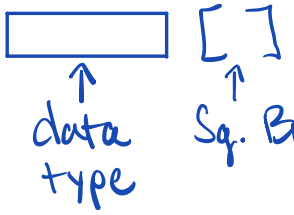
Use switch and break for user input to choose options of iteration

Category of data types

Primitive - start with lowercase letter

Reference - All else; start with capital letter




ex: int[] arr;

ref. to array obj.

Value is null initially

arr = new int[size];


Integer[] arr1;

arr1 = new Integer[size];

| # items | # bits needed |
|---|---|
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 3 |
| 8 | 3 |
| 9 | 9 |

1 byte = 8 bits

| # bits | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # items | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 1MB | 1G |

$2^{10}$ = 1Kb

4 bytes 2 bytes 8 bytes

int, short (poor cousin), long (wealthy cousin)

4 bytes 2 bytes

double, float

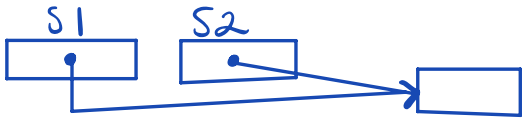
Java statements

- declaration
- assignment
- conditional
- loop
- method call
- return statement
- exception

Static data fields are accessible before instance of Class exists

Java has a String pool for constant strings

```
S1 = "Hello";
S2 = "Hello";
```




Identifiers

- variables — scope: Method it was declared
- method names - Scope: Class it's in
- class names - Scope:

Memory Allocation

- dynamic memory allocation → using new

Pointer in c++ is a reference in java which is a memory address

```
int x = . . . ;
int y = . . . ;

int max;

if (x>y) {
   max = x;
}

else {
   max = y;
}
```

Ternary operator

```
max = ((x>y) ? x : y);
```

Errors

- compilation
- Runtime
- Logic
- Programming Style

Loops:

For, while, do-while

Abstract Data Types → data structure
collection of data plus functionality → array

- insert
- remove
- retrieve

# LAB 2 stuff

List: linear collection that supports insert, remove, retrieve for any valid position

0 size-1
7 9 -3 12 9

remove/retrieve [0, size-1]

Stack: linear collection _____ only one end

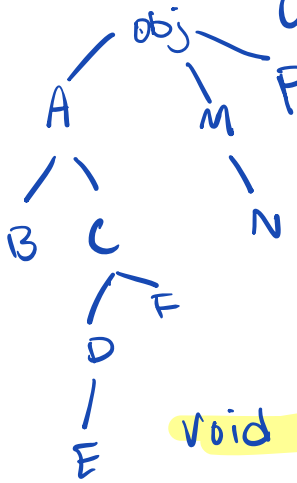
Queue: linear _____ insert ops at opposite end of remove/retrieve supported

DEQ _____ insert/retrieve/remove

List ADT Interface

or push for add
or pop remove

```
boolean isEmpty();
int size();
void add(Object item, int index) throws exception   -> out of bounds
Object remove(int index)          - throws exception
Object.get(int index)             - throws exception
```




A a; Static wise a is of obj A
a = new D(); dynamic wise a is of obj D

a.toString(); calls D's .toString()

a.instanceOf( bool

void removeAll()

create Array → items