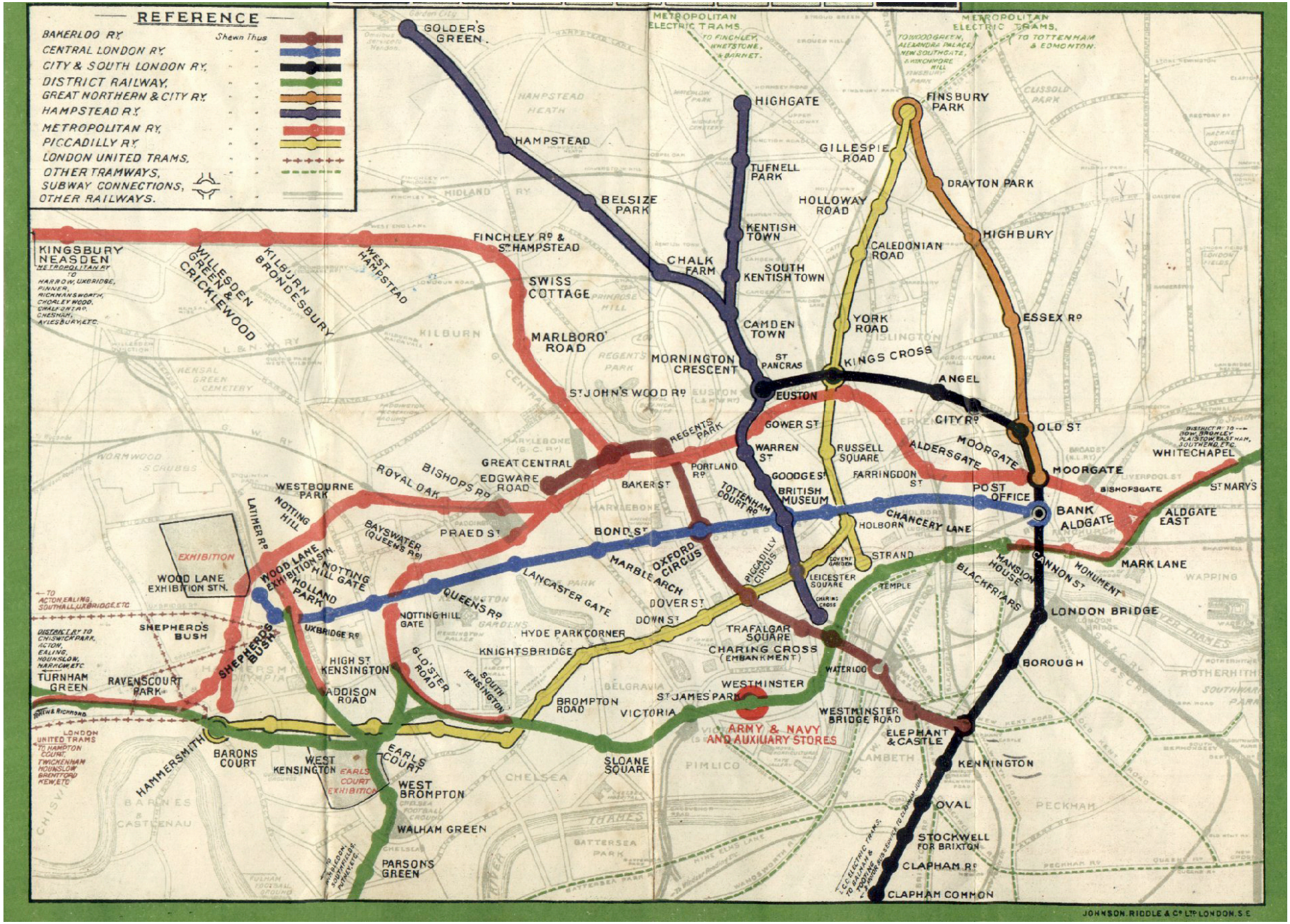
Object-Oriented Programming


REFERENCE
Shewn Thus
BAKERLOO RY.
CENTRAL LONDON RY.
CITY & SOUTH LONDON RY.
DISTRICT RAILWAY.
GREAT NORTHERN & CITY RY.
HAMPSTEAD RY.
METROPOLITAN RY.
PICCADILLY RY.
LONDON UNITED TRAMS.
OTHER TRAMWAYS.
SUBWAY CONNECTIONS.
OTHER RAILWAYS.
GOLDER'S GREEN.
HAMPSTEAD
BELSIZE PARK
HIGHGATE
FINSBURY PARK
GILLESPIE ROAD
TUFNELL PARK
DRAYTON PARK
HOLLOWAY ROAD
KENTISH TOWN
HIGHBURY
CALEDONIAN ROAD
CHALK FARM
SOUTH KENTISH TOWN
KINGSBURY NEASDEN
WILLESDEN GREEN & CRICKLEWOOD
KILBURN BRONDESBURY
WEST HAMPSTEAD
FINCHLEY Rd & St. HAMPSTEAD
SWISS COTTAGE
MARLBORO' ROAD
CAMDEN TOWN
YORK ROAD
ESSEX Rd
MORNINGTON CRESCENT
ST PANCRAS
KINGS CROSS
ANGEL
EUSTON
ST JOHN'S WOOD Rd
GOWER ST
CITY Rd
OLD ST
REGENTS PARK
WARREN ST
RUSSELL SQUARE
ALDERSGATE
MOORGATE
WHITECHAPEL
GREAT CENTRAL
EDGWARE ROAD
PORTLAND Rd
GOODGE St
FARRINGDON ST
BISHOPS Rd
ROYAL OAK
BAKER ST
BRITISH MUSEUM
POST OFFICE
BISHOPSGATE
St MARY'S
WESTBOURNE PARK
NOTTING HILL
LATIMER Rd
TOTTENHAM COURT Rd
CHANCERY LANE
BANK
ALDGATE
ALDGATE EAST
BAYSWATER (QUEENS Rd)
PRAED ST
BOND ST
HOLBORN
EXHIBITION
WOOD LANE EXHIBITION STN.
WOOD LANE EXHIBITION STN.
NOTTING HILL GATE
HOLLAND PARK
OXFORD CIRCUS
PICCADILLY CIRCUS
STRAND
MANSION HOUSE
CANNON ST
MONUMENT
MARK LANE
MARBLE ARCH
LANCASTER GATE
QUEENS Rd
LEICESTER SQUARE
TEMPLE
BLACKFRIARS
LONDON BRIDGE
SHEPHERD'S BUSH
NOTTING HILL GATE
DOVER ST
DOWN ST
HYDE PARK CORNER
TRAFALGAR SQUARE
CHARING CROSS (EMBANKMENT)
TURNHAM GREEN
RAVENSCOURT PARK
HIGH ST KENSINGTON
GLOUCESTER ROAD
SOUTH KENSINGTON
KNIGHTSBRIGHTSBRIDGE
WATERLOO
BOROUGH
ADDISON ROAD
BROMPTON ROAD
ST JAMES PARK
WESTMINSTER
VICTORIA
WESTMINSTER BRIDGE ROAD
ARMY & NAVY AND AUXILIARY STORES
ELEPHANT & CASTLE
KENNINGTON
HAMMERSMITH
BARONS COURT
WEST KENSINGTON
EARLS COURT
EARLS COURT EXHIBITION
WEST BROMPTON
SLOANE SQUARE
OVAL
WALHAM GREEN
STOCKWELL FOR BRIXTON
PARSONS GREEN
CLAPHAM Rd
CLAPHAM COMMON
METROPOLITAN ELECTRIC TRAMS. TO FINCHLEY, WHETSTONE, & BARNET.
METROPOLITAN ELECTRIC TRAMS. TO WOODGREEN, ALEXANDRA PALACE, NEW SOUTHGATE, & WINCHMORE HILL
METROPOLITAN ELECTRIC TRAMS. TO TOTTENHAM & EDMONTON.
JOHNSON, RIDDLE & Co Ltd LONDON, S.E.

1908

Harry Beck, 1933

Abstraction

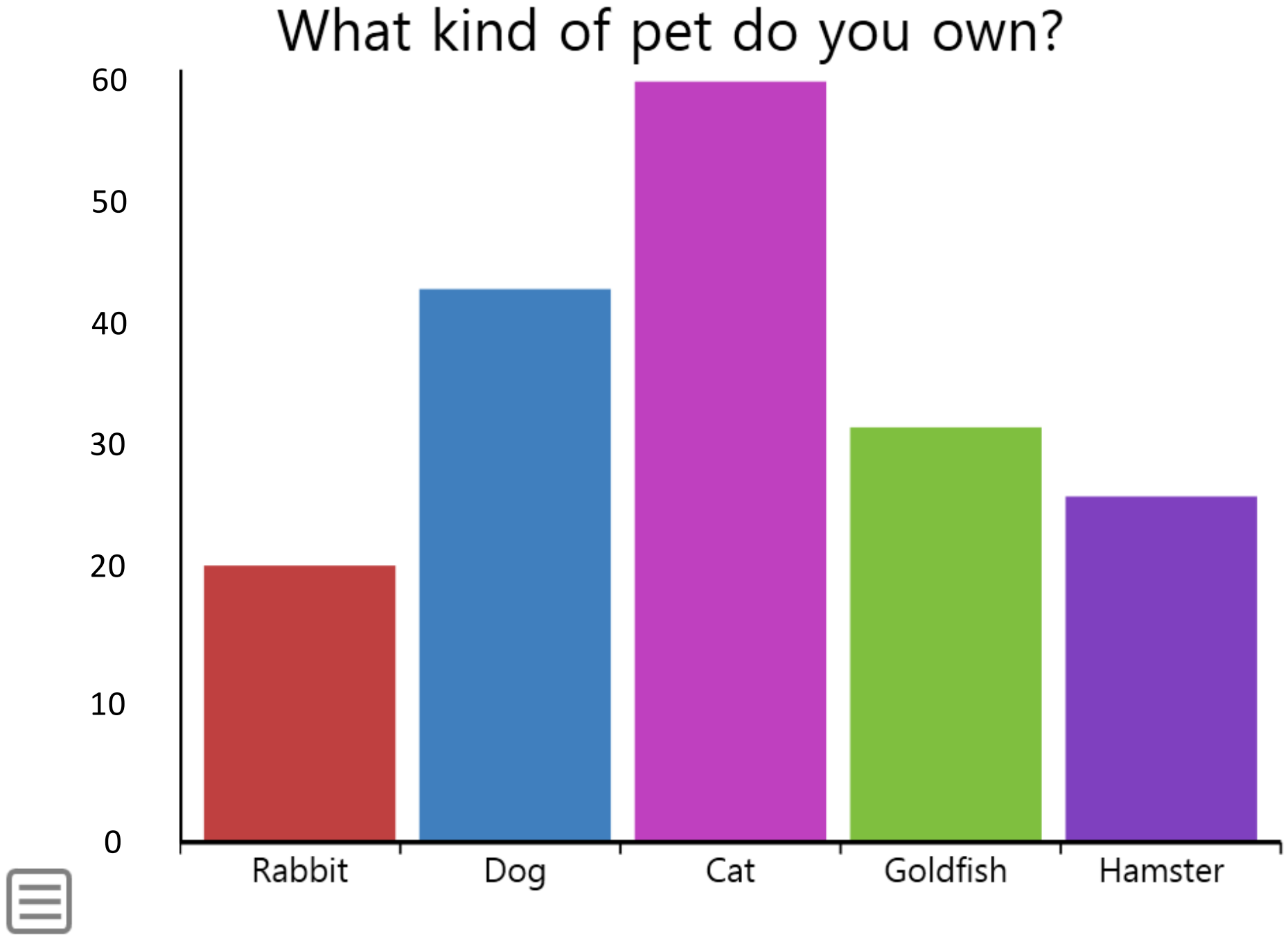
- Abstraction is the process of capturing only those ideas about a concept that are relevant to the current situation.
- Irrelevant ideas are left out.

Abstraction

- ***Control abstraction***: Giving function names to sections of code that then "stand" for that code.
- When we call a function, we don't care how the function works, we just care that it does work.
  - We have captured the meaning of a section of code by giving it a name, while giving the caller of the function the ability to ignore how it works.

Abstraction

- ***Data abstraction***: Choosing to represent a concept by including certain features and ignoring others.


What kind of pet do you own?
60
50
40
30
20
10
0
Rabbit
Dog
Cat
Goldfish
Hamster

Classes

- Classes = Structs + Functions
- A class is a struct with some functions associated with it that act upon that struct.
- The point of a class is to combine data abstractions (a struct) with appropriate control abstractions (functions), resulting in one entity that has ***state*** (variables) and associated ***behaviors*** (functions).

Design a class


The SIMS

Two questions

- When designing a class, answer:
  - What properties or attributes do instances of my class possess?
    - These become the ***fields*** (data members) of your class.
    - These are usually nouns.
  - What actions can instances of my class do?
    - These become the ***methods*** (member functions) of your class.
    - These are usually verbs.
  - Collectively, fields and methods are the ***members*** of your class.

```
class name_of_class {
  public:
    type field1;
    type field2;          // more fields...
    type method1(...);
    type method2(...);    // more methods...
  private:
    type field1;
    type field2;          // more fields...
    type method1(...);
    type method2(...);    // more methods...
};
```

Designing a class

- Classes are declared like structs, but have public and private sections.
- Anything in the public section is accessible by programmers ***writing*** or ***using*** the class.
- Anything in the private section is accessible only by the programmer ***writing*** the class.
- (More about this later.)

- Add a method called `play()` that shows the dog playing. A dog can't play unless its energy is positive. This method works just like `bark()` except it also depletes one unit of energy when the dog plays.
- Add a method called `eat(int howmuch)` that shows the dog eating. The dog recovers "howmuch" units of energy when this method is called.