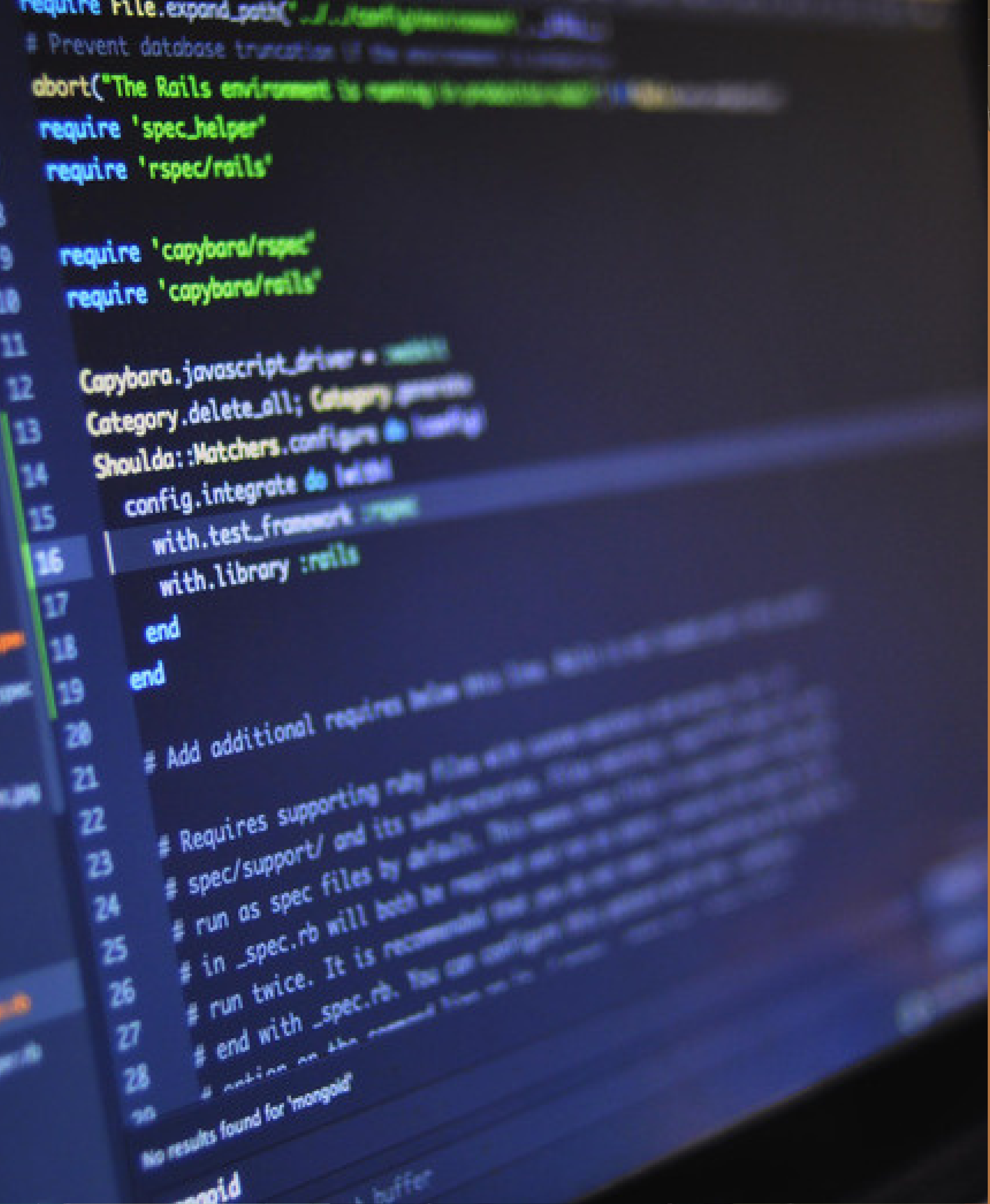
Placements 2024 batch

Trust yourself, Keep Practicing



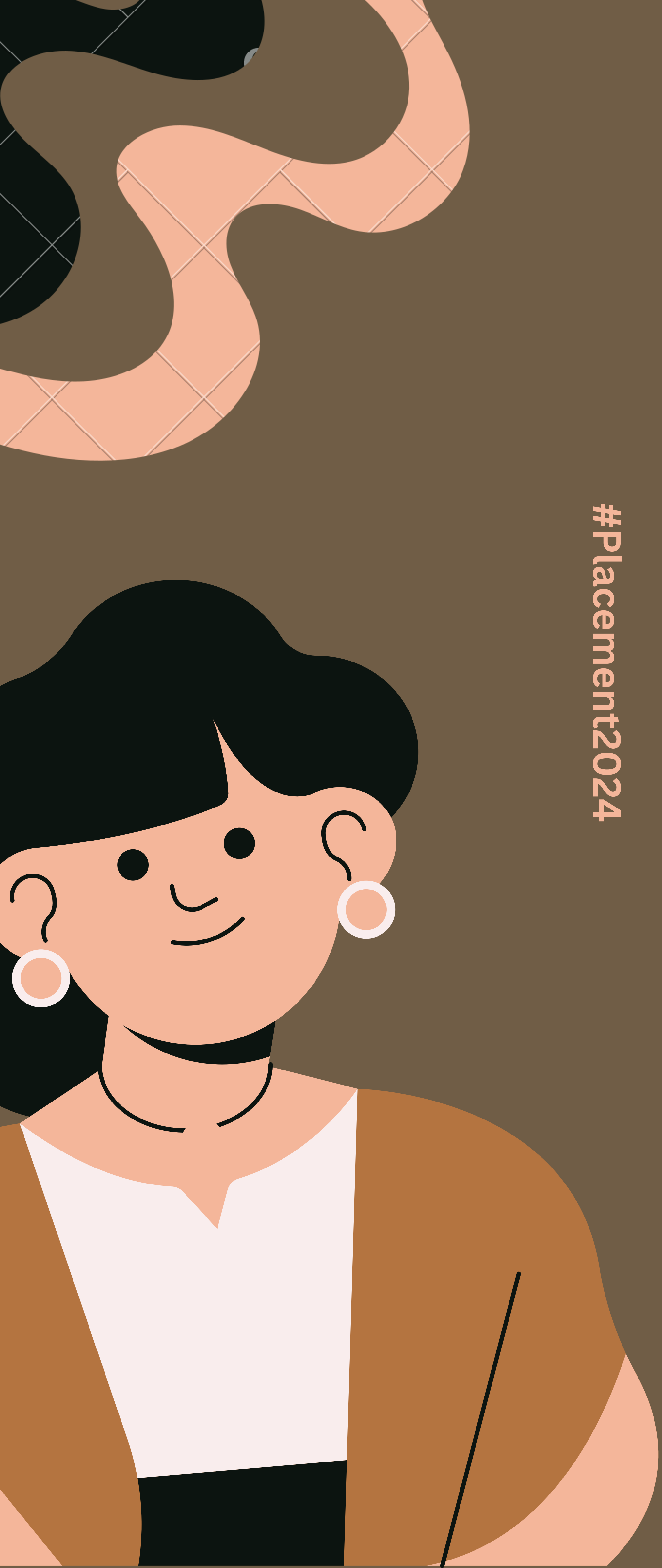
```
require File.expand_path('../../[illegible]
# Prevent database truncation [illegible]
abort("The Rails environment [illegible]
require 'spec_helper'
require 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver = [illegible]
Category.delete_all; Category.[illegible]
Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end

# Add additional requires below this line. [illegible]

# Requires supporting ruby files with [illegible]
# spec/support/ and its subdirectories. [illegible]
# run as spec files by default. [illegible]
# in _spec.rb will both be required [illegible]
# run twice. It is recommended [illegible]
# end with _spec.rb. You can configure [illegible]
# option on the [illegible]
```

No results found for 'mongoid'

STEP 1

MASTER DATA STRUCTURES

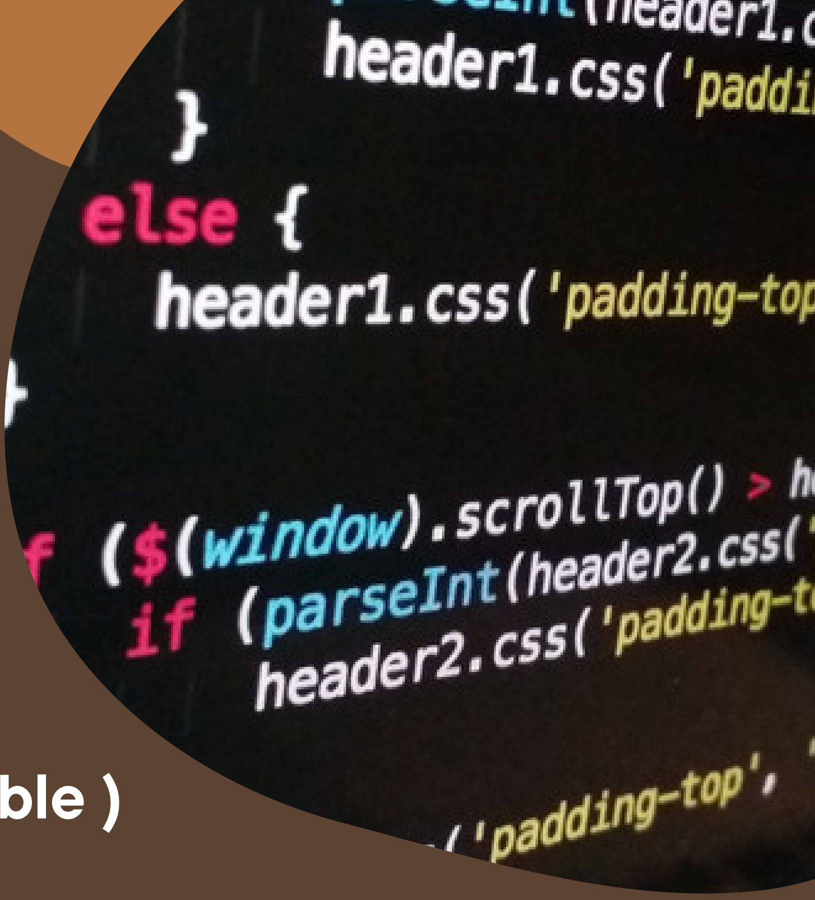
Data structures & Algorithms



#Placement2024

1. **Quality Over Quantity:** Solving numerous problems isn't enough. Focus on diverse question types, patterns, and scenarios for true understanding.
2. **Implement from Scratch:** Go beyond templates. Learn to build data structures and solutions from the ground up.
3. **Beyond Code Sheets:** Mastering concepts matters. Test your logic in contests and challenges for real problem-solving skills.

#Placement2024



```
header1.css('paddi
}
else {
  header1.css('padding-top
}
f ($(window).scrollTop() > h
  if (parseInt(header2.css(
    header2.css('padding-t
('padding-top',
```

DATA STRUCTURES

1. Array
2. String
3. Stack
4. Queue
5. Dequeue
6. Linked List (Single and Double)
7. Heaps
8. HashMap and HashSet
9. Tries
10. Trees
11. Graphs
12. Union Find

ALGORITHMS

1. Sorting: Merge Sort, Quick Sort, Radix Sort, Counting Sort
2. Searching: Linear Search, Binary Search
3. Graph Traversal: DFS, BFS
4. Tree Traversal: Preorder, Postorder, Inorder and Morris Traversal
5. Dijkstra, Prim’s and Kruskal
6. Kosaraju and Tarjan’s algorithm for strongly connected components
7. Sieve of Eratosthenes
8. Floyd’s cycle detection algorithm

STEP 2

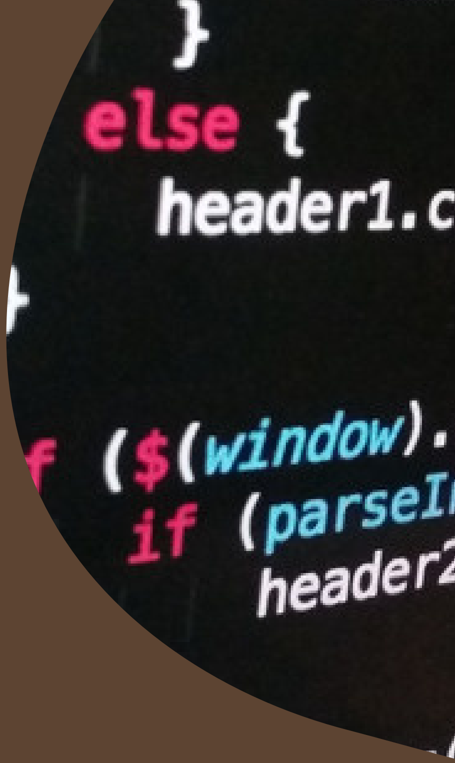
MASTER APTITUDE

Aptitude, English Proficiency

1. **Understand, Don't Memorize:** Grasp formulas by understanding their application and tricks.
2. **Prioritize Accuracy:** Focus on accuracy before speed. Practice extensively to enhance both.
3. **Daily Mixed Practice:** Engage in daily mixed sets for comprehensive thinking and concept application.

#Placement2024

#Placement2024



```
   }
else {
   header1.c
}
f ($(window).
   if (parseI
      header2
```

GROUP 1
1. **Numbers**
2. **HCF or LCM**
3. **Fractions & Decimals**
4. **Surds & Indices**
5. **Simplification**
6. **Square Root & Cuber Root**

GROUP 2
1. **Average**
2. **Problem on Ages**

GROUP 3
1. **Speed, Distance, Time**
2. **Boats, Streams**
3. **Problem on Trains**

GROUP 4
1. **Time and Work**
2. **Pipes and Cisterns**

GROUP 5
1. **Percentage**
2. **Ratio and Propotion**
3. **Work and Wages**
4. **Chain Rule**
5. **Partnership**
6. **Allegation and Mixture**

GROUP 6
1. **Discount**
2. **Profit and Loss**

GROUP 7
1. **Simple Interest**
2. **Compound Interest**

GROUP 8
1. **Probability**
2. **Permutations & Combinations**

GROUP 9
1. **Clock**
2. **Calendar**

GROUP 10
1. **Area**
2. **Volume and Surface Area**

GROUP 11
1. **Height and Distance**

STEP 3

MASTER CORE SUBJECTS

Core Subjects

#Placement2024

1. **Object-Oriented Mastery:** Choose a language (Java recommended) and delve into Inheritance, Polymorphism, Abstraction, Encapsulation, Classes, Objects.
2. **Database Proficiency:** Grasp SQL languages, syntax, aggregate functions, JOINS, subqueries, and stored procedures.
3. **OS Expertise:** Master Scheduling, Deadlock, Locks, Semaphores, Multi-Threading, Processes, and Threads for Operating Systems.

Remember, placement is a journey, not a race. Prioritize self-improvement over stress and competition. Focus on your preparation and growth.

Thank you!