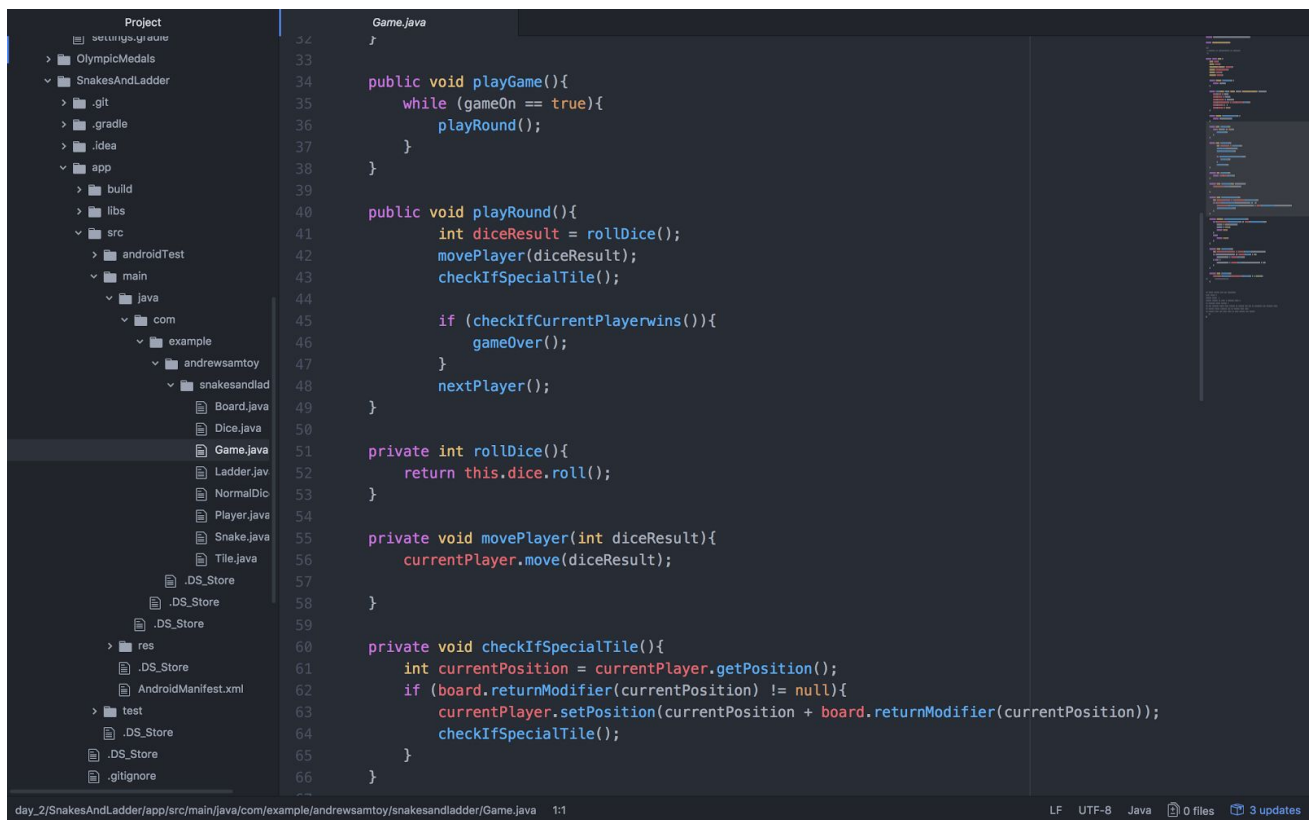


I.T. 1 - Encapsulation

Take a screenshot of an example of encapsulation in a program.



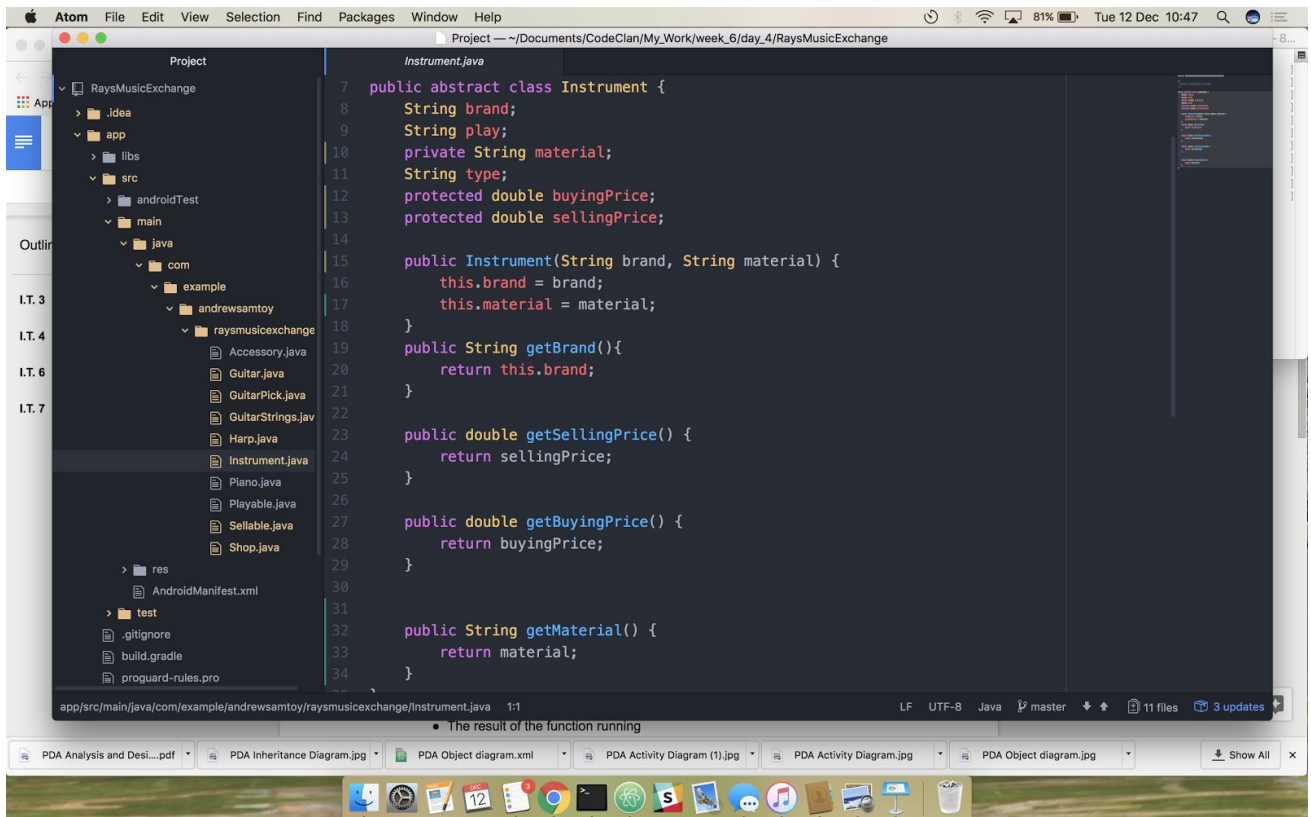
The screenshot shows an IDE with a project structure on the left and the `Game.java` file open in the center. The project structure includes a `src/main/java/com/example/andrewsamtoy/snakesandladder` package containing `Board.java`, `Dice.java`, `Game.java`, `Ladder.java`, `NormalDice.java`, `Player.java`, `Snake.java`, and `Tile.java`. The `Game.java` file contains the following code:

```
32 }
33
34 public void playGame(){
35     while (gameOn == true){
36         playRound();
37     }
38 }
39
40 public void playRound(){
41     int diceResult = rollDice();
42     movePlayer(diceResult);
43     checkIfSpecialTile();
44
45     if (checkIfCurrentPlayerwins()){
46         gameOver();
47     }
48     nextPlayer();
49 }
50
51 private int rollDice(){
52     return this.dice.roll();
53 }
54
55 private void movePlayer(int diceResult){
56     currentPlayer.move(diceResult);
57 }
58
59
60 private void checkIfSpecialTile(){
61     int currentPosition = currentPlayer.getPosition();
62     if (board.returnModifier(currentPosition) != null){
63         currentPlayer.setPosition(currentPosition + board.returnModifier(currentPosition));
64         checkIfSpecialTile();
65     }
66 }
```

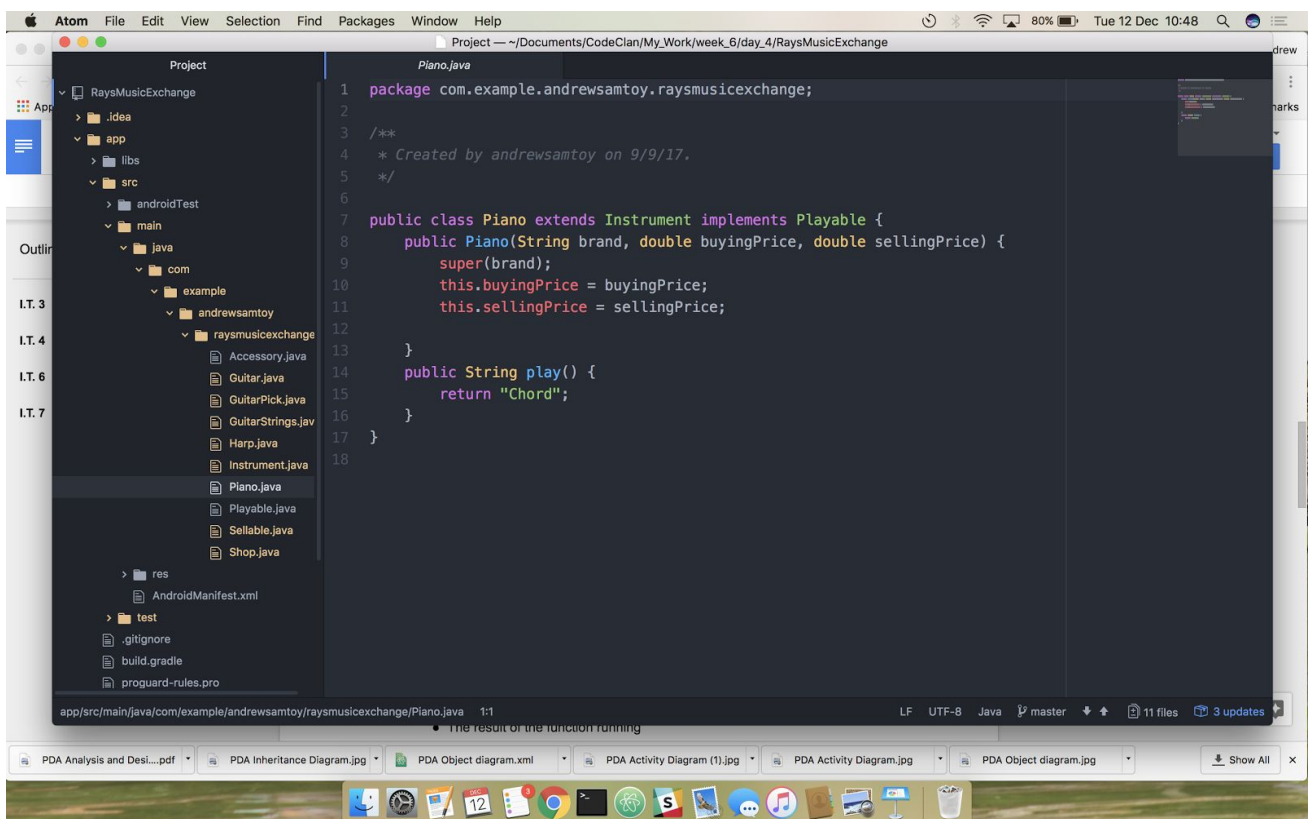
I.T. 2 - Inheritance in a program

Take a screenshot of the use of Inheritance in a program. Take screenshots of:

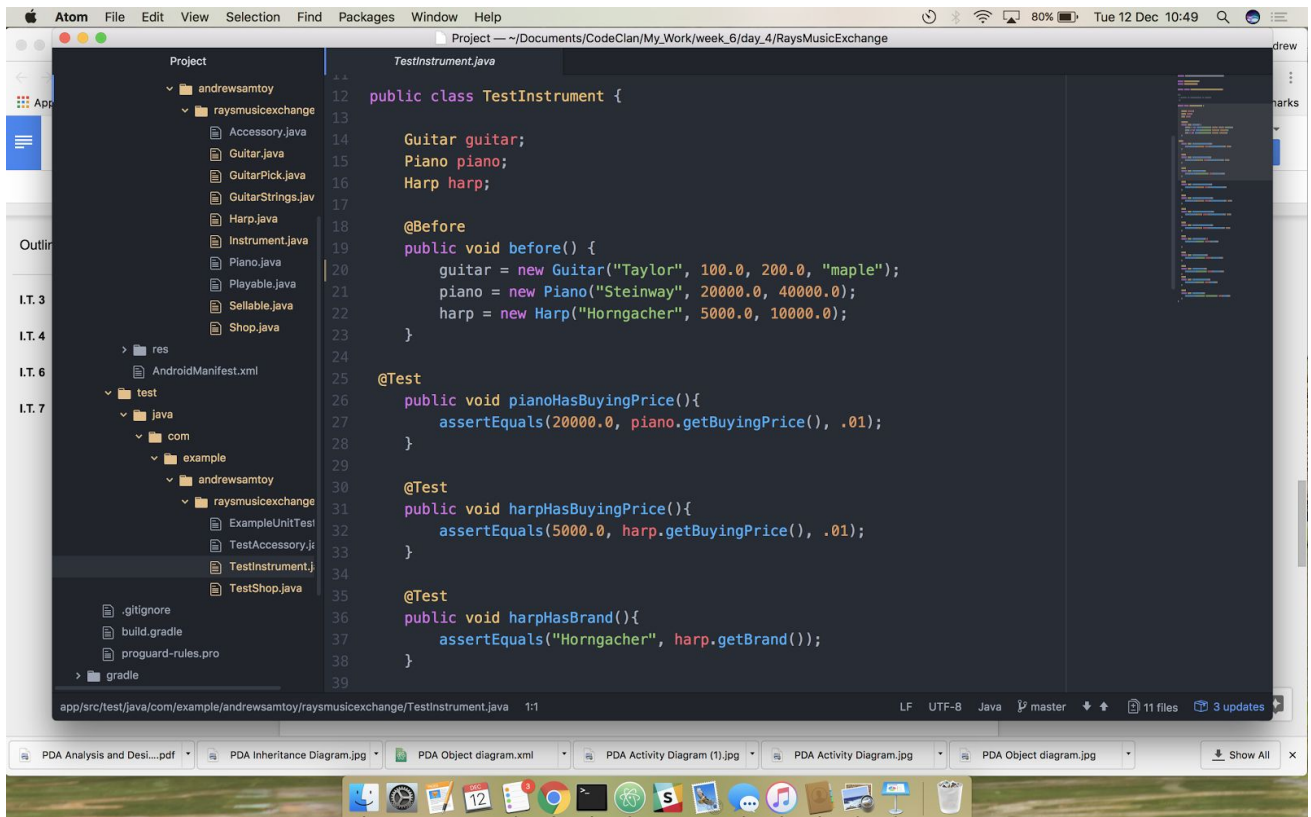
- A Class



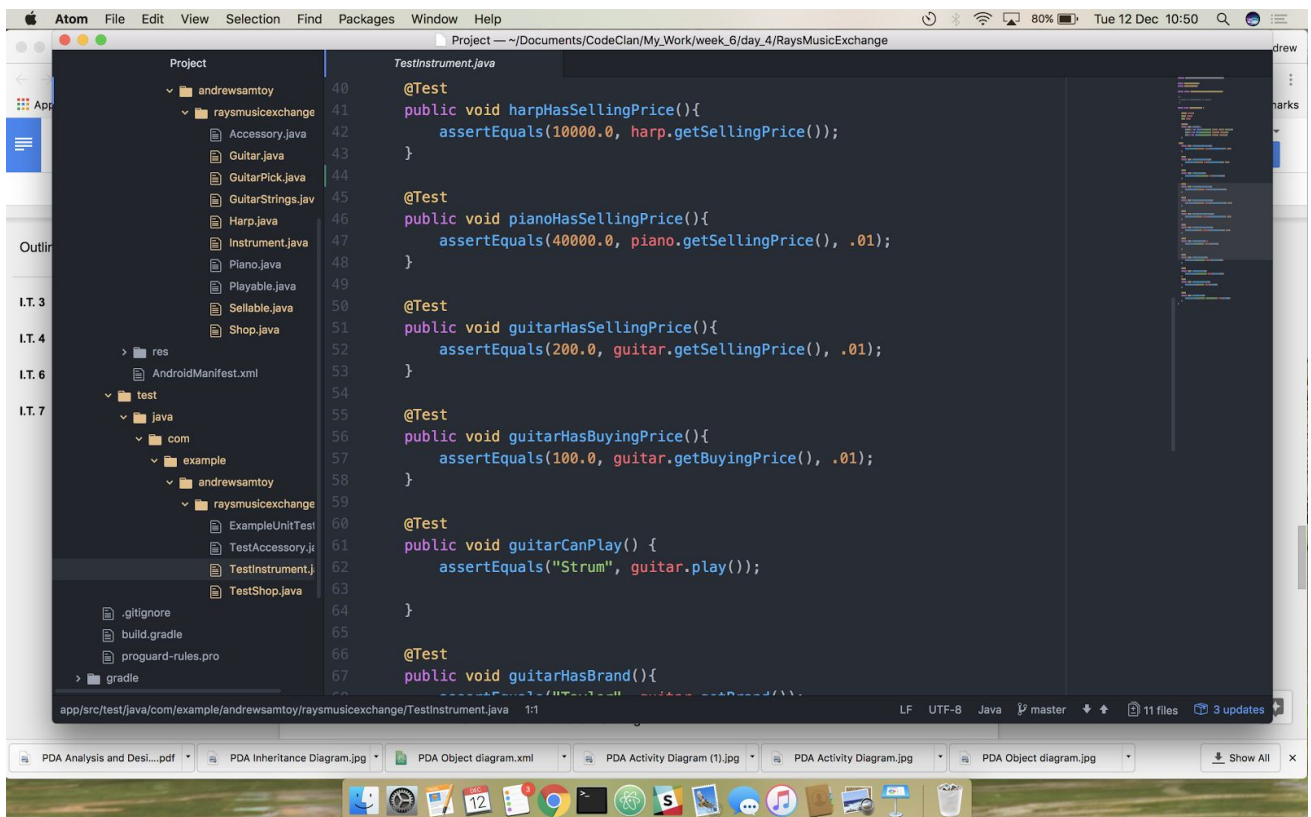
- A Class that inherits from the previous class



- An Object in the inherited class



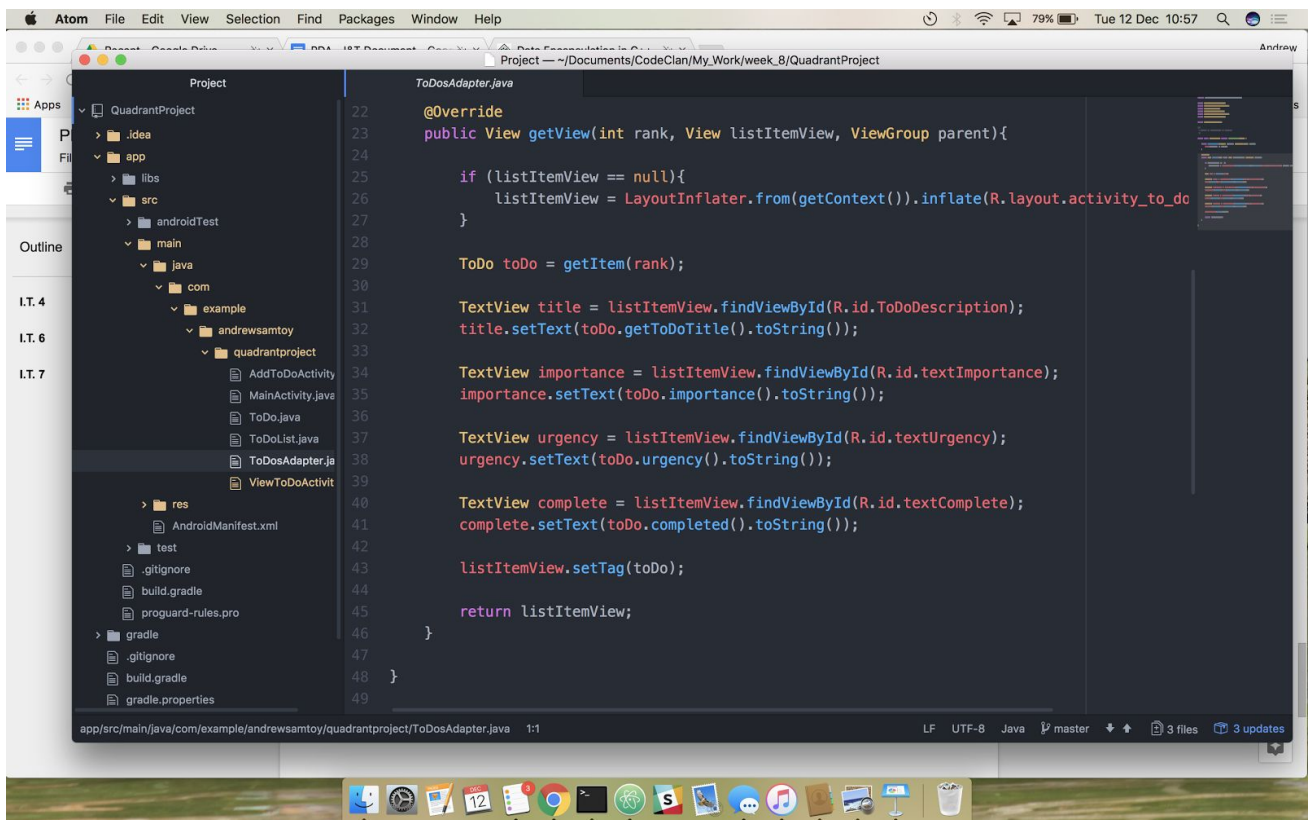
- A Method that uses the information inherited from another class.



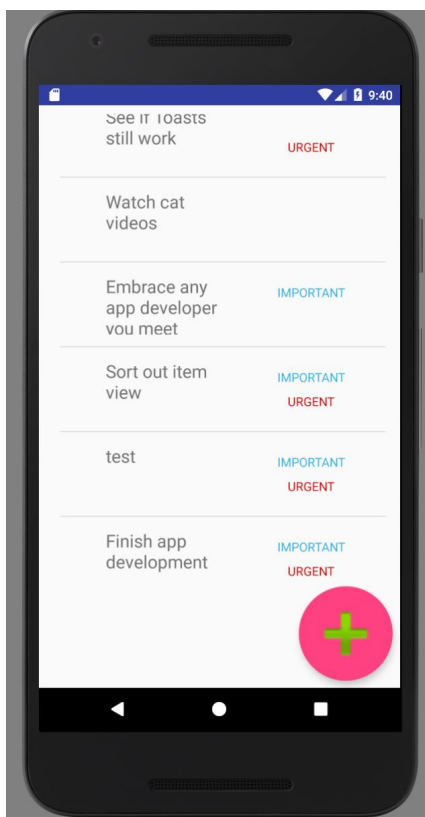
I.T. 3

Demonstrate searching data in a program. Take screenshots of:

- Function that searches data



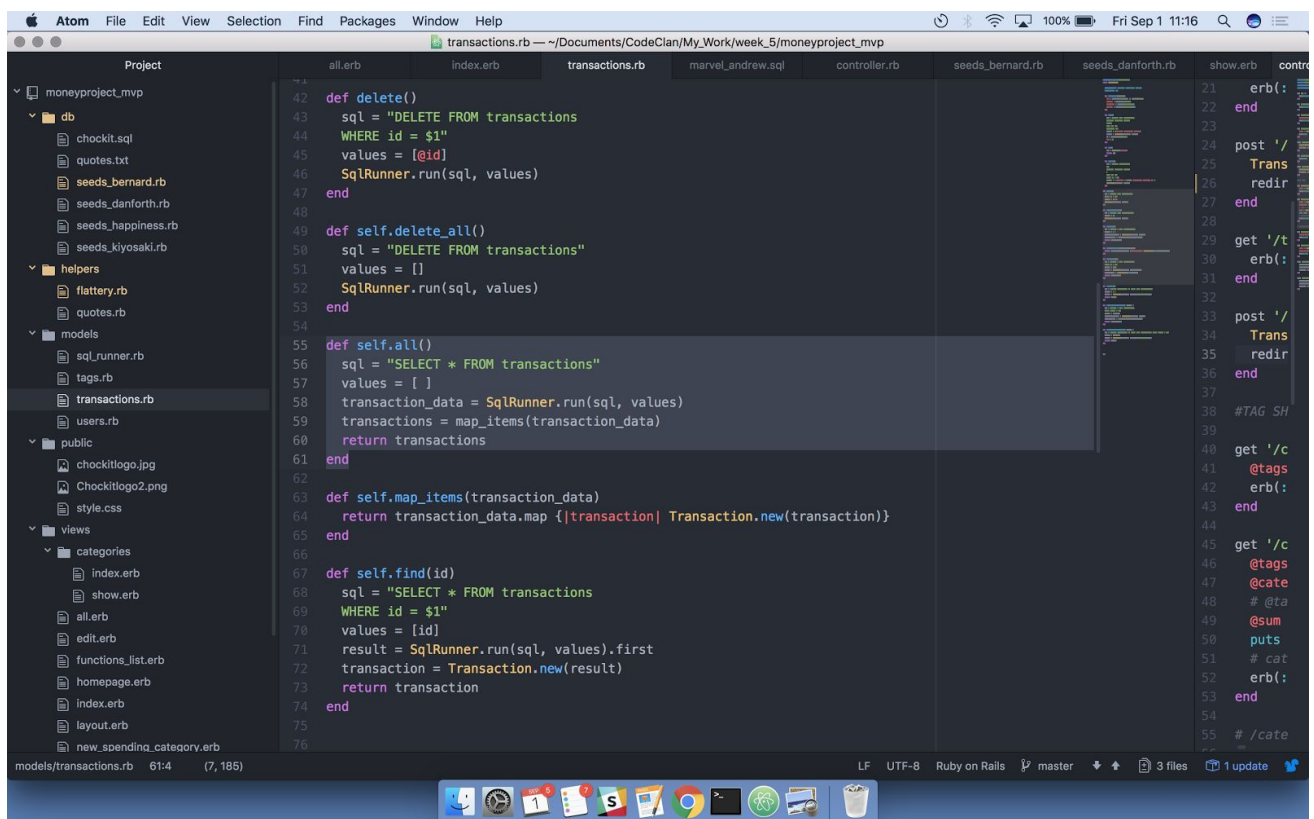
- The result of the function running



I.T. 4

Demonstrate sorting data in a program. Take screenshots of:

- Function that sorts data



```
def delete()
  sql = "DELETE FROM transactions
  WHERE id = $1"
  values = [@id]
  SqlRunner.run(sql, values)
end

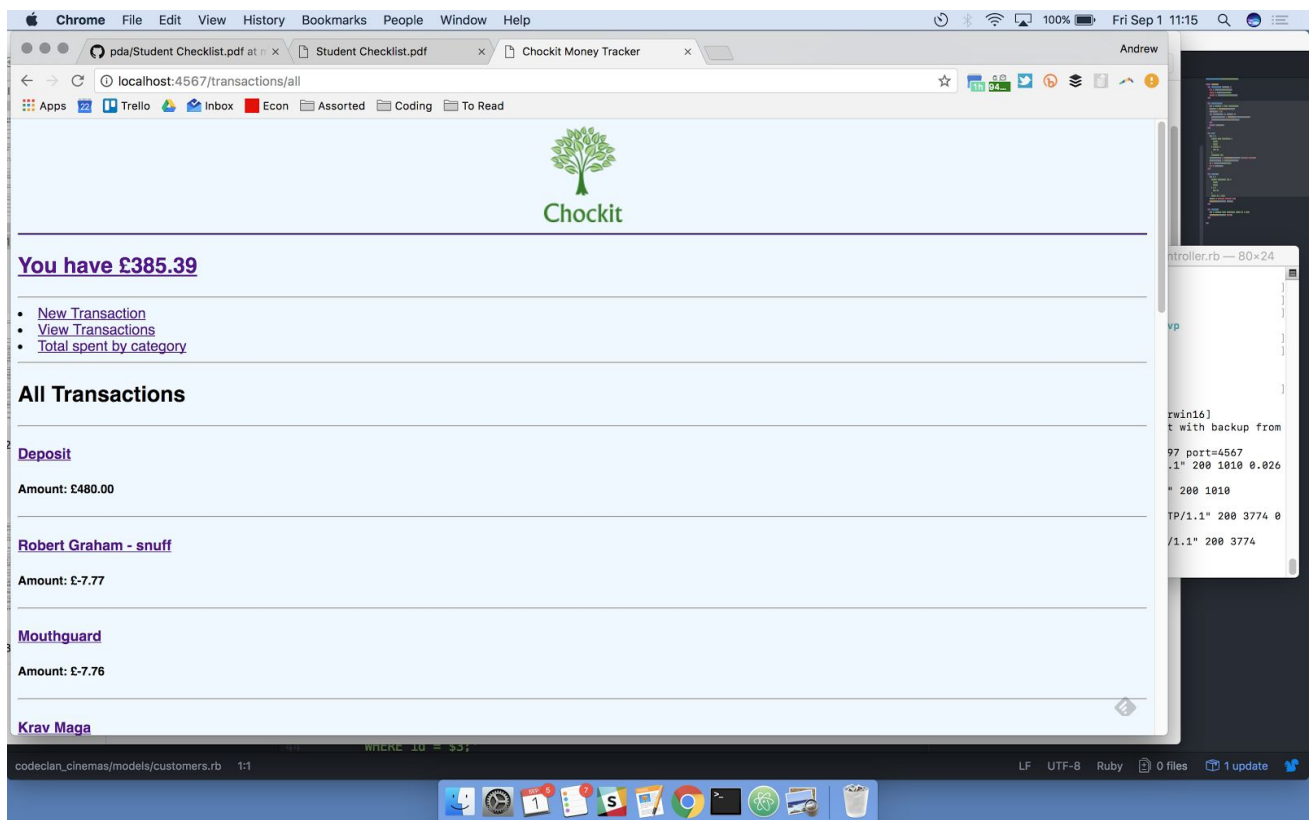
def self.delete_all()
  sql = "DELETE FROM transactions"
  values = []
  SqlRunner.run(sql, values)
end

def self.all()
  sql = "SELECT * FROM transactions"
  values = [ ]
  transaction_data = SqlRunner.run(sql, values)
  transactions = map_items(transaction_data)
  return transactions
end

def self.map_items(transaction_data)
  return transaction_data.map {|transaction| Transaction.new(transaction)}
end

def self.find(id)
  sql = "SELECT * FROM transactions
  WHERE id = $1"
  values = [id]
  result = SqlRunner.run(sql, values).first
  transaction = Transaction.new(result)
  return transaction
end
```

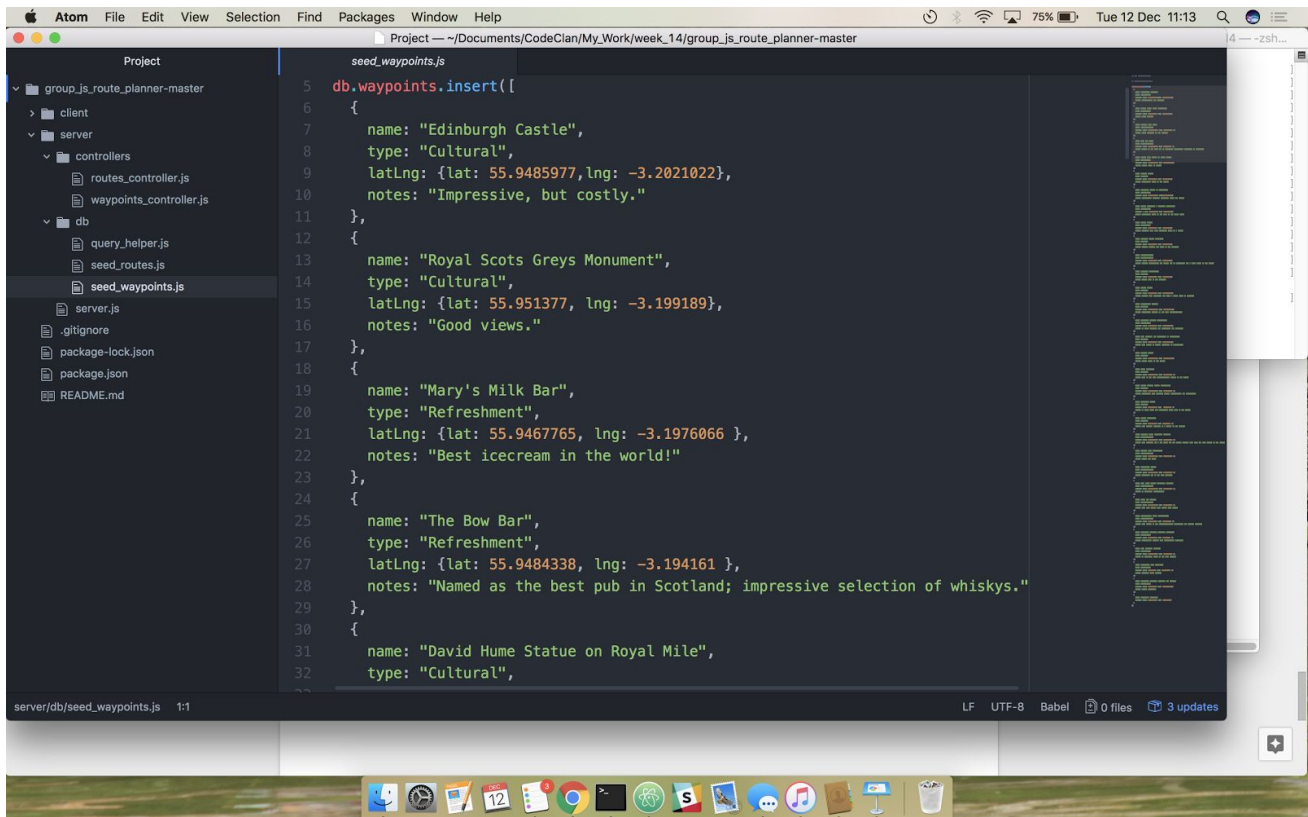
- The result of the function running



I.T. 5

Demonstrate the use of an array in a program. Take screenshots of:

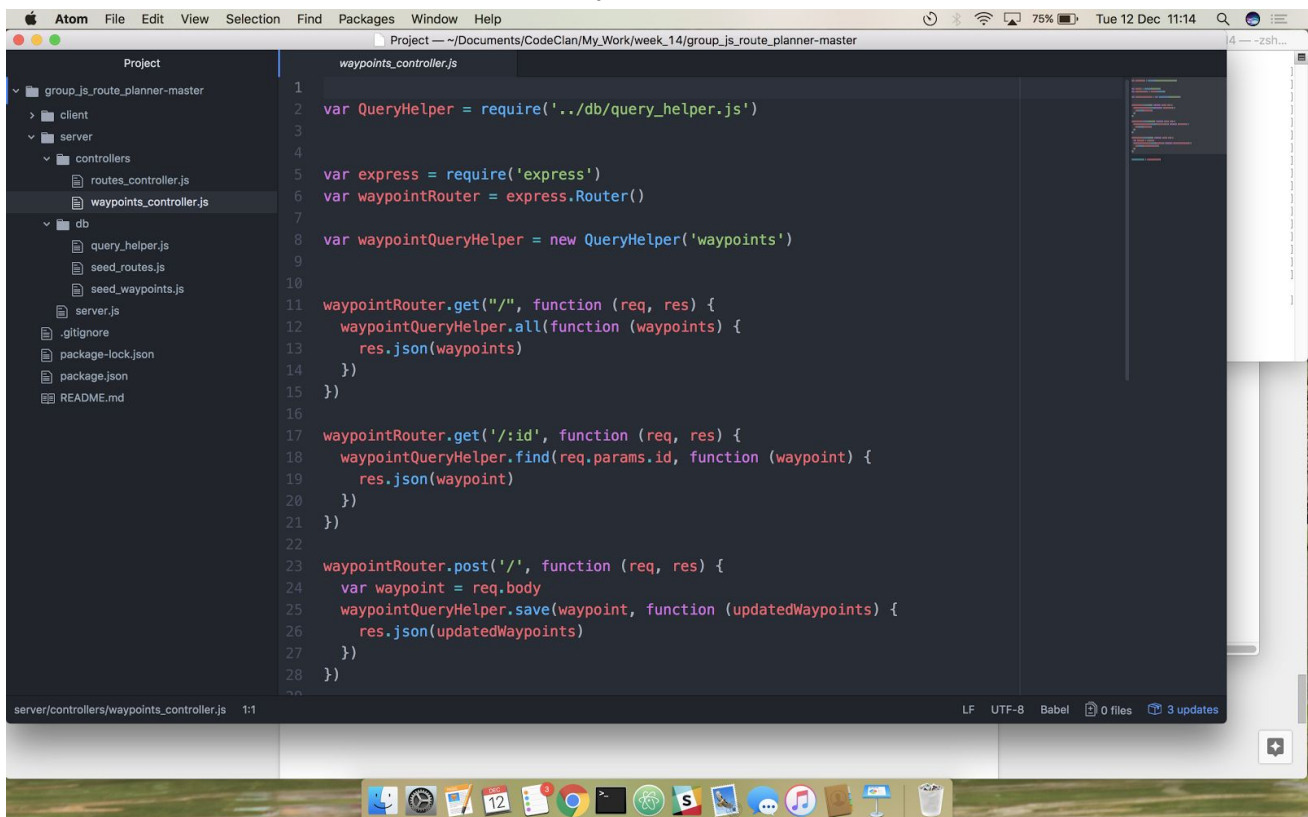
- An array in a program



The screenshot shows the Atom editor interface with a project named 'group_js_route_planner-master'. The file explorer on the left shows the project structure, including 'client', 'server', 'controllers', 'db', and 'server.js'. The main editor window displays the file 'seed_waypoints.js' with the following code:

```
1 db.waypoints.insert([
2   {
3     name: "Edinburgh Castle",
4     type: "Cultural",
5     latLng: {lat: 55.9485977, lng: -3.2021022},
6     notes: "Impressive, but costly."
7   },
8   {
9     name: "Royal Scots Greys Monument",
10    type: "Cultural",
11    latLng: {lat: 55.951377, lng: -3.199189},
12    notes: "Good views."
13  },
14  {
15    name: "Mary's Milk Bar",
16    type: "Refreshment",
17    latLng: {lat: 55.9467765, lng: -3.1976066 },
18    notes: "Best icecream in the world!"
19  },
20  {
21    name: "The Bow Bar",
22    type: "Refreshment",
23    latLng: {lat: 55.9484338, lng: -3.194161 },
24    notes: "Named as the best pub in Scotland; impressive selection of whiskys."
25  },
26  {
27    name: "David Hume Statue on Royal Mile",
28    type: "Cultural",
29  },
30 ])
```

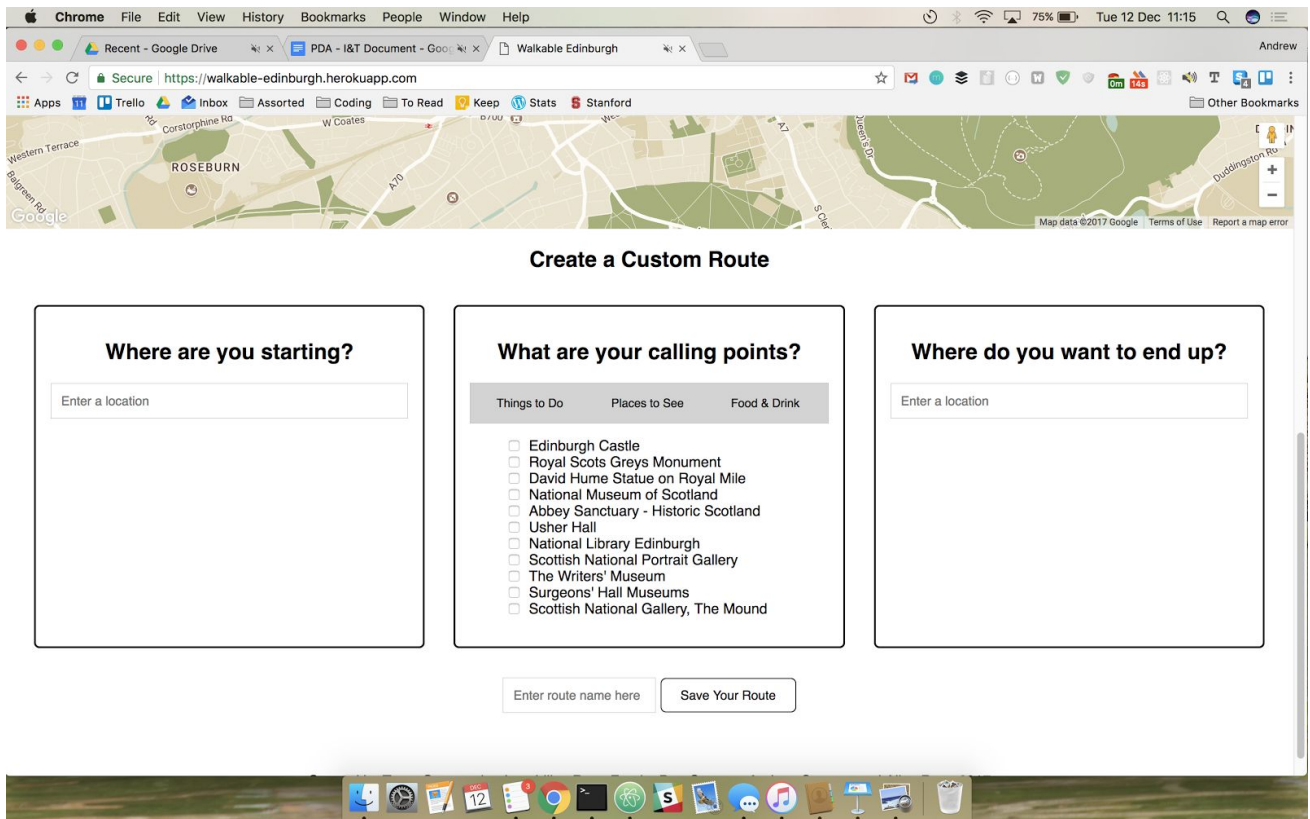
- A function that uses the array



The screenshot shows the Atom editor interface with the same project. The file explorer on the left shows the project structure, including 'client', 'server', 'controllers', 'db', and 'server.js'. The main editor window displays the file 'waypoints_controller.js' with the following code:

```
1
2 var QueryHelper = require('../db/query_helper.js')
3
4
5 var express = require('express')
6 var waypointRouter = express.Router()
7
8 var waypointQueryHelper = new QueryHelper('waypoints')
9
10
11 waypointRouter.get('/', function (req, res) {
12   waypointQueryHelper.all(function (waypoints) {
13     res.json(waypoints)
14   })
15 })
16
17 waypointRouter.get('/:id', function (req, res) {
18   waypointQueryHelper.find(req.params.id, function (waypoint) {
19     res.json(waypoint)
20   })
21 })
22
23 waypointRouter.post('/', function (req, res) {
24   var waypoint = req.body
25   waypointQueryHelper.save(waypoint, function (updatedWaypoints) {
26     res.json(updatedWaypoints)
27   })
28 })
```

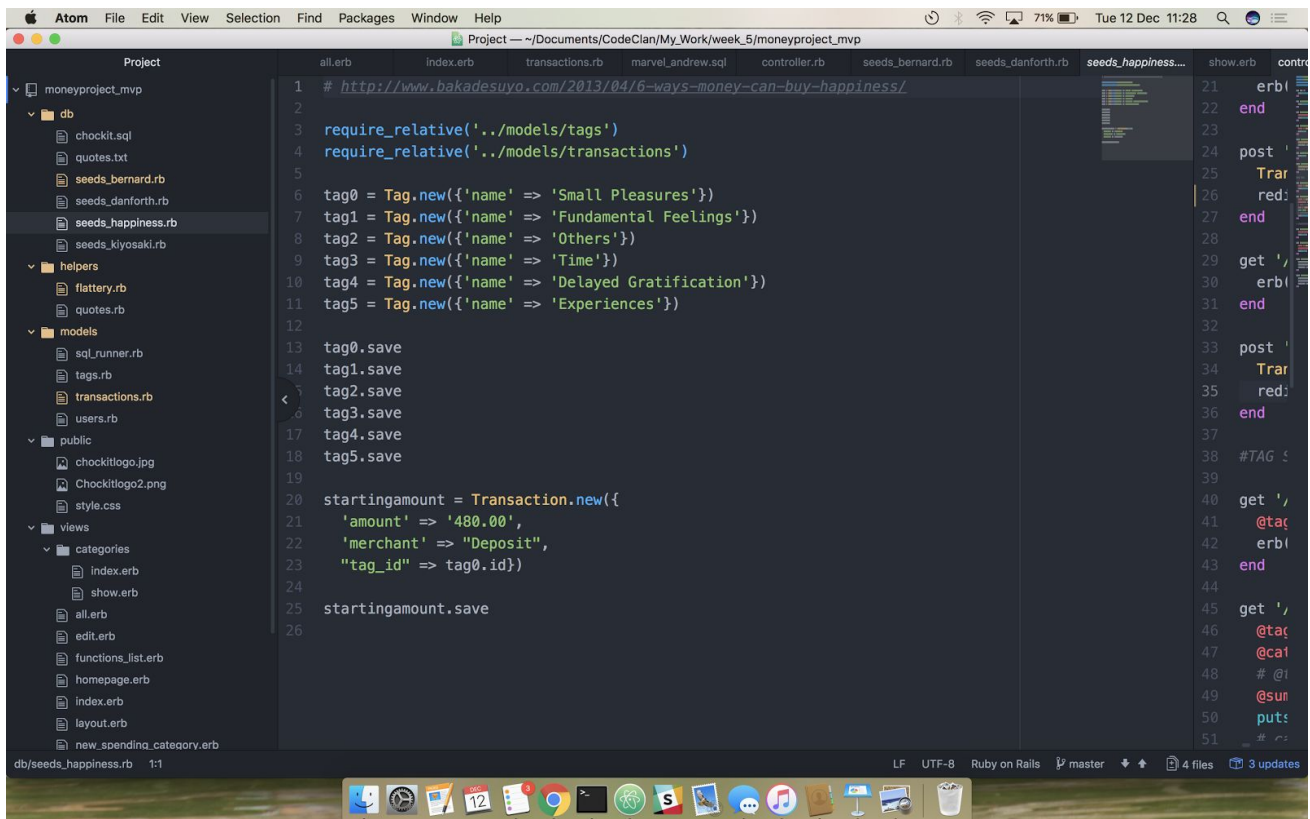
- The result of the function running



I.T. 6

Demonstrate the use of a hash in a program. Take screenshots of:

- A hash in a program



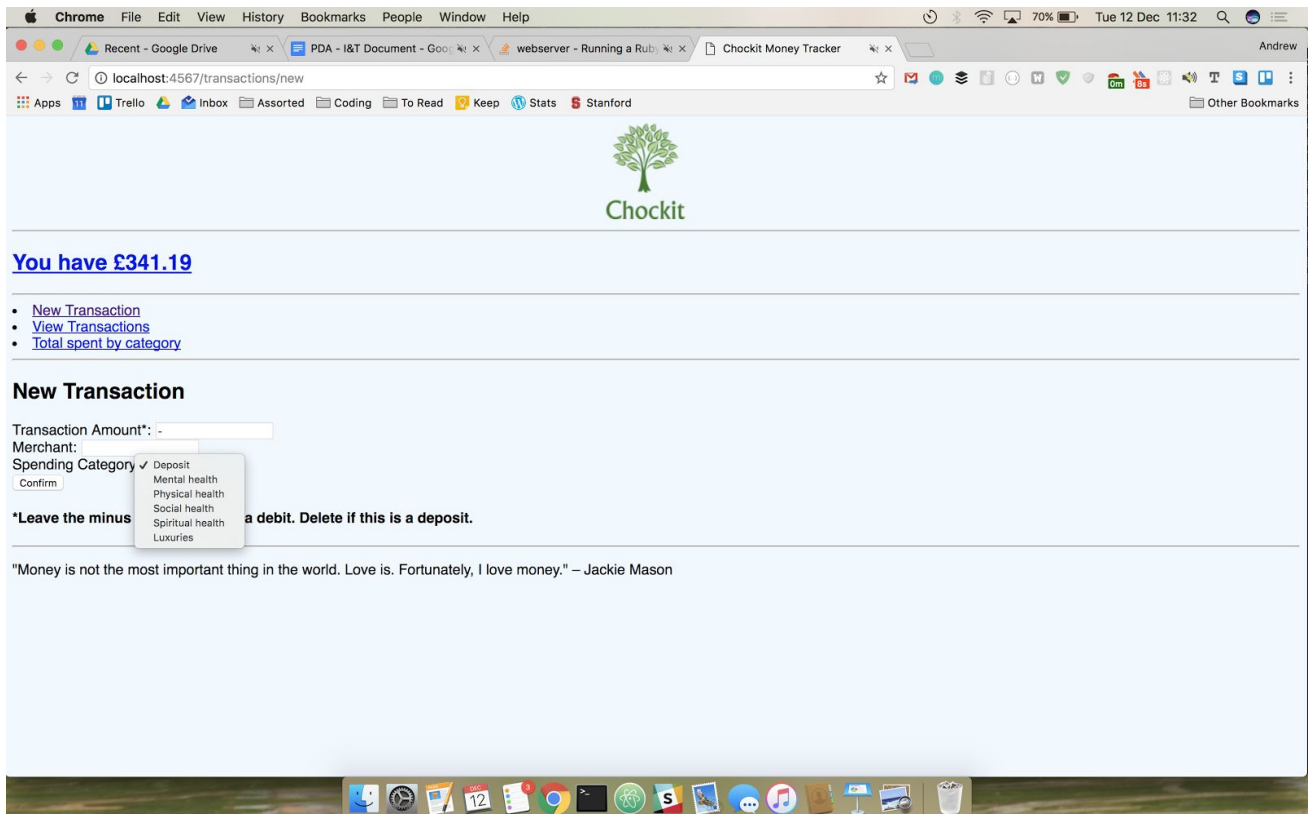
- A function that uses the hash


```
Atom  File  Edit  View  Selection  Find  Packages  Window  Help
Project — ~/Documents/CodeClan/My_Work/week_5/moneyproject_mvp
all.erb  index.erb  transactions.rb  new.erb  marvel_andrew.sql  controller.rb  seeds_bernard.rb  seeds_danforth.rb  show.erb  contr

1 <h2>New Transaction</h2>
2 <form method="POST" action="/transactions">
3   <div class="form-wrapper">
4
5     <div class="form-trans">
6       <label for="amount">Transaction Amount*:</label>
7       <input type="numeric" name="amount" id="amount" value="-"/>
8     </div>
9
10    <div class="form-trans">
11      <label for="merchant">Merchant:</label>
12      <input type="text" name="merchant" id="merchant"/>
13    </div>
14
15    <div class="form-trans">
16      <label for="tag">Spending Category:</label>
17      <select name="tag_id" id="tag_select">
18        <% @tags.each do |tag| %>
19          <option value=<%= tag.id %>><%= tag.name.capitalize %></option>
20        <% end %>
21      </select>
22    </div>
23
24    <div class="form-trans">
25      <input type="submit" value="Confirm"/>
26    </div>
27  </div>
28
29 </form>
30
31 <h4>*Leave the minus sign in if this is a debit. Delete if this is a deposit.</h4>

21 erb
22 end
23
24 post '
25   Trar
26   red:
27 end
28
29 get '/'
30 erb
31 end
32
33 post '
34   Trar
35   red:
36 end
37
38 #TAG
39
40 get '/'
41   @ta
42   erb
43 end
44
45 get '/'
46   @ta
47   @ca
48   # @
49   @su
50   put
51   # r
```

• The result of the function running



I.T. 7

Demonstrate the use of Polymorphism in a program.