Quiz - 1

Create Multi Fill Question | Create Multiple Choice Question

Create Descriptive Question | Create Parsons Problem Question

---

Question: Can you find anything wrong with the code on the board?

Answer: Will be discussed in class

Edit Question

```
class Print
  def print_owing
        outstanding = 0.0

        # print banner
        puts “*************************”
        puts “**Customer Owes ***”
        puts “*************************”

        #calculate outstanding
        @orders.each do |order|
              outstanding += order.amount
        end

        #print details
        puts(“name: #{@name})
        puts(“amount: #{outstanding})

        puts “*************************”
        puts “**Customer Owes ***”
        puts “*************************”
   end
end
```

Answer: Long method and duplicate code

Quiz 2: Refactor Code (/tests/2)

Create Multi Fill Question | Create Multiple Choice Question | Create Descriptive Question | Create Parsons Problem Question

---

Question: Given code on the board, using extract method the code has been refactored. Arrange the following code blocks to match the refactored solution.

Drag from here

```
balance = previousAmount * 1.2
```

```
end
```

```
print_banner
balance = get_balance(previousAmount * 1.2)
print_details(balance)
print_banner
```

```
end
```

```
balance = initial_value
  @orders.each do |order|
    balance += order.amount
  end
```

```
#calculate balance
```

```
balance
```

```
def tuition_owed(previousAmount)
```

```
def print_details(balance)
  puts("name: #{_name}")
  puts("amount: #{balance}")
```

Construct your solution here

```
end
```

```
#print details
```

```
def get_balance initial_value
```

```
def print_banner
  puts "************************"
  puts "**University of Memphis ***"
  puts "************************"
end
```

```
# print banner
```

Reset | Get feedback

Edit Question

Answer:

```
class Print
  def print_owing
        printBanner
        outstanding = get_outstanding()
        print_details(outstanding)
        printBanner
  end
  def get_outstanding          #calculate outstanding
        outstanding = 0.0
        @orders.each do |order|
              outstanding += order.amount
        end
        outstanding
  end

  def print_details(amount)            #print details
        puts("name: #{_name}")
        puts("amount: #{amount}")
 end

 def print_banner                        # print banner
        puts "*************************"
        puts "**Customer Owes ***"
        puts "*************************"
 end
end
```

Quiz 3: Smells and Refactorings (/tests/3)

Create Multi Fill Question | Create Multiple Choice Question | Create Descriptive Question | Create Parsons Problem Question

---

Question 1

Which of the following refactorings is best suited for resolving duplicate code and long methods?

Option 1: Move Method

Option 2: Extract Method

Option 3: Remove Parameter

Option 4: Hide Method

Answer: Extract Method

Edit Question

Question 2

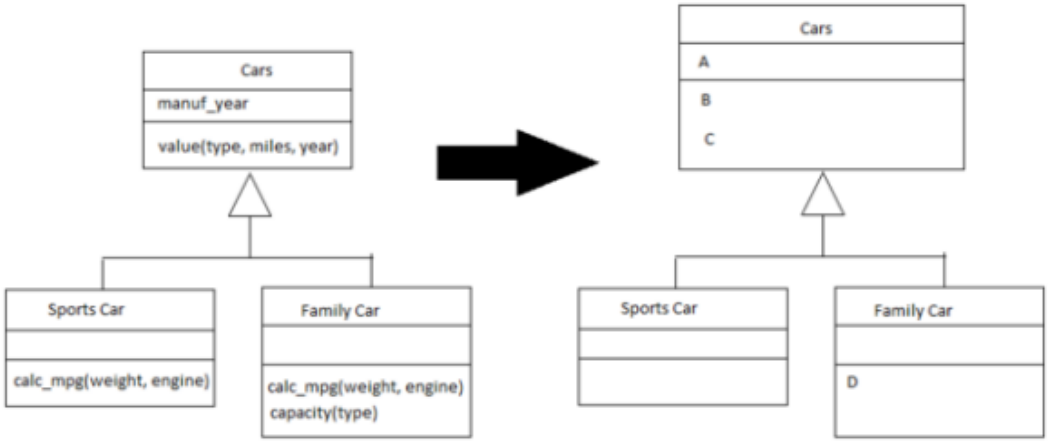
The diagram, reflects pull up method being applied, fill in the corresponding diagram by selecting the labels from the drop-down.

A:

B:

C:

D:



Answer:

A: manuf_year

B: value(type, miles, year)

C: calc_mpg(weight, engine)

D: capacity(type)

Edit Question

Question 3

The code on the right is extract method applied to the code fragment on the left. Fill in the missing pieces.

A: ▾

B: ▾

C: ▾

D: ▾

```
def foo
    a = 0
    b = 1
    c = 2
    return a * b  + c * 123
end
```

```
A
    B
end
C
    b = 1
    a = 0
    c = 2
    D
end
```

Answer:

A: def bar(a, b, c)

B: return a * b + c * 123

C: def foo

D: return bar(a, b, c)

Edit Question