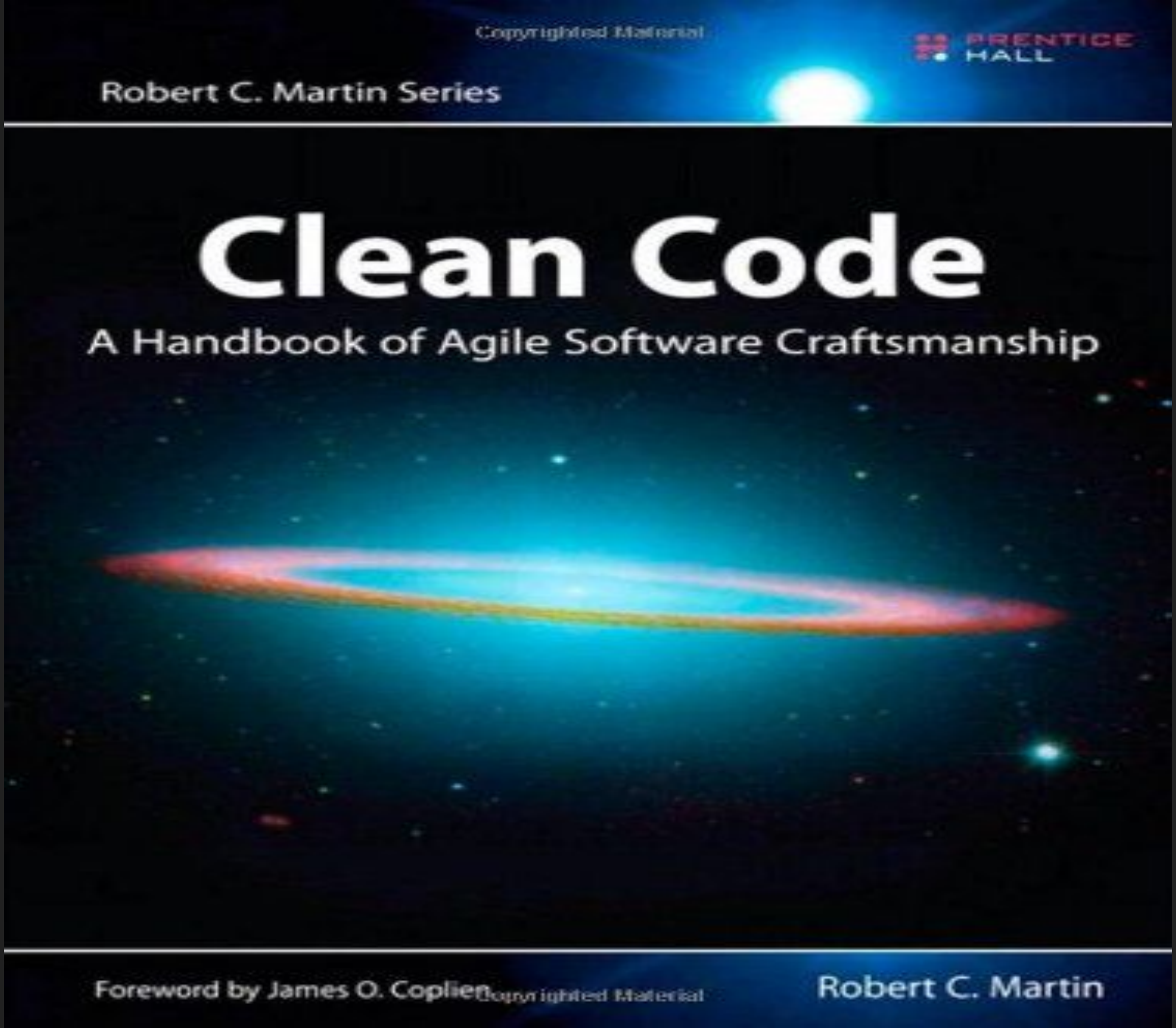
```
def bulk_update(self, objs, fields, batch_size=None):
        """
        Update the given fields in each of the given objects in the database.
        """
        if batch_size is not None and batch_size < 0:
            raise ValueError('Batch size must be a positive integer.')
        if not fields:
            raise ValueError('Field names must be given to bulk_update().')
        objs = tuple(objs)
        if any(obj.pk is None for obj in objs):
            raise ValueError('All bulk_update() objects must have a primary key set.')
        fields = [self.model._meta.get_field(name) for name in fields]
        if any(not f.concrete or f.many_to_many for f in fields):
            raise ValueError('bulk_update() can only be used with concrete fields.')
        if any(f.primary_key for f in fields):
            raise ValueError('bulk_update() cannot be used with primary key fields.')
        if not objs:
            return
        # PK is used twice in the resulting update query, once in the filter
        # and once in the WHEN. Each field will also have one CAST.
        max_batch_size = connections[self.db].ops.bulk_batch_size(['pk', 'pk'] + fields,
        batch_size = min(batch_size, max_batch_size) if batch_size else max_batch_size
        requires_casting = connections[self.db].features.requires_casted_case_in_updates
        batches = (objs[i:i + batch_size] for i in range(0, len(objs), batch_size))
        updates = []
        for batch_objs in batches:
            update_kwargs = {}
            for field in fields:
                when_statements = []
```



PRENTICE HALL

Robert C. Martin Series

Clean Code

A Handbook of Agile Software Craftsmanship

Foreword by James O. Coplien

Robert C. Martin

What is Clean Code

- Always Looks Like It Was Written By Someone Who Cares. (Michael Feathers)

- If You Want Your Code To Be Easy To Write Make It Easy To Read (Uncle Bob)

Is Code That Is Easy To Understand & Easy To Change

Naming

1- use intention revealing names like (book ,author,...) not (x ,y) .

```
 $X = [5,4] , $y = [‘ali’,’salah’]
$numbers= [5,4]  , $names = [‘ali’,’salah’]

 fname( $str1 , $str2 ) ….. getFullName ( $firstName , $lastName)
```

(You aren’t writing for yourself , you also write for other developers)

Naming

2- if you put a comment on a variable name then in most cases the name doesn't reveal its intention

```
$m = $x + $y  #m is the sum of two  products prices
```

change it to

```
$totalPrice = $firstProductPrice + $secondProductPrice
```

Naming

3- you can use i or x as a variable for inner loops , but also try to be explicit not implicit , for python principles type (import this) in python shell

```
foreach $L1 as $i
        #do something

#this is more readable and better understanding

foreach $books as $book
        # do something
```

Functions (Small)

1- small , small ,small , if you make scroll in a functions then you are (عاكك)
try to make max numbers of lines between 20 to 30 lines .

```
function search ($name) {
      If ($name == ‘mohamed’) {
           $ticketPrice = 30
           $result = queryDatabase($tickectPrice, $name)
           If ($result == 0) {
              $flagResult =setFlagInDatabase (1)
              If ($flagResult == 1) {
                 # and keep scrolling to the end of function
```

X

Functions (Small)

2- make function do one thing only like this example

```
function printPage () {
      echo "<html> <head>
                  <link rel="stylesheet" type="text/css" href="theme.css">
                  </head>
             <body>
                  <h1> hello world </h1>
                  <p> this is a test paragraph for the first html page training </p>
            <script> alert('page is loaded'); </script>
           <script> alert('now i am loading jquery'); </script>
         <script> alert('enjoy surfing the page'); </script>
       </body>
            <footer><p>copyright for: Hege Refsnes Contact information:
             <a href="http://google.com">Visit Me</a>.</p>
      </footer></html>"
```

Functions (Small)

previous function contains too many responsibilities and also it's ugly and have much lines of code , let's try to refactor it

```
function header () {
        return "<html>
                         <head>
                         <link rel="stylesheet" type="text/css" href="theme.css">
                        </head> "

function footer () {
                return "<footer>
                                <p>copyright Hege Refsnes Contact information:
                                <a href="http://google.com">Visit Me</a>.
                                </p>
                        </footer>
                         </html>"
```

Functions (Small)

```
function body () {
    return "<body>
            <h1> hello world </h1>
            <p> this is a test for the first html page training </p>
            <script> alert('page is loaded'); </script>
            <script> alert('now i am loading jquery'); </script>
            <script> alert('enjoy surfing the page'); </script>
            </body>"


function printPage () {
            return print header() . body() . footer()
```

Functions (Small)

3- avoid inner loops & long line length

```
    $email = ‘ahmed    @        yahoo    . com ‘
```

X

```
If ( trim($email) == ‘ ’ or trim($email)  == null or trim($email)  == false or …..):
```

A better solution

```
function isValidEmail ($email):
    return preg_match(‘some patterns ‘ , $email)
```

Functions (Small)

```
$matrix = [ [‘one’,’two’] , [‘three’,’four’] ]

    foreach $matrix as $row :
        foreach $row as $column :
            echo $column
                        #outputs  one
                                  two
```

X

A better solution #source from
(http://stackoverflow.com/questions/1821471/python-nested-loop)

```
    foreach $matrix as $row :
        echo implode(‘\n’, $row)
```

Summary

1- use intention revealing naming

2- write the code in minimum lines

3- make functions small as much as possible

4- function should be responsible for doing one thing

5- avoid inner loops and long line length

6- it’s ok if you are (عاكك) but make refactoring as soon as possible and don’t accumulate technical debts