QUESTION 1 — 1 points — Save Answer

What is a draw back of the Mediator pattern?

- ○ A. It creates lots of little objects that all look alike.
- ○ B. It hides implementation details from client.
- ○ C. It complicates object protocols.
- ○ D. It couples colleagues.
- ○ E. It centralises control.

QUESTION 2 — 1 points — Save Answer

The Mediator pattern couples colleagues.

- ○ True
- ○ False

QUESTION 3 — 1 points — Save Answer

The Mediator pattern is classified as:

- ○ Creational
- ○ Behavioural
- ○ Structural

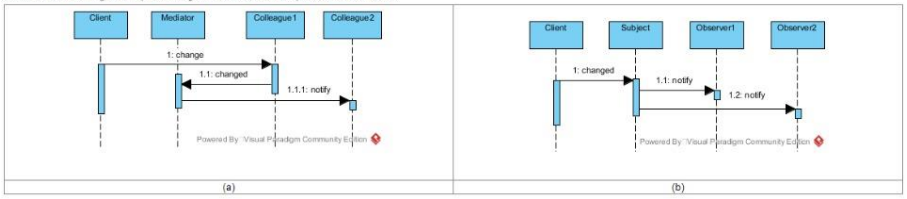
QUESTION 4 — 1 points — Save Answer

Which pattern observes an object and reacts appropriately when the state of the object changes?

- ○ Iterator
- ○ Mediator
- ○ Memento
- ○ Observer

QUESTION 5 — 6 points — Save Answer

Consider the following two sequence diagrams and answer the questions which follow.



Note: *You may assume that the diagrams include only object names and that the corresponding class names are the same as the object names without the numbers.*
Hint: *Do not read meaning into the names of the functions on the diagrams. These function names do not correspond to those with the same names in the pattern.*

1. Identify the design patterns represented by each of the sequence diagrams. (2)
   (a)
   (b)
2. Which participant(s) in (a) defined the `notify()` function? (1)
3. Which participant(s) in (b) defined the `notify()` function? (1)

Note: *You may assume that the diagrams include only object names and that the corresponding class names are the same as the object names without the numbers.*
Hint: *Do not read meaning into the names of the functions on the diagrams. These function names do not correspond to those with the same names in the pattern.*

1. Identify the design patterns represented by each of the sequence diagrams. (2)
   (a) Mediator
   (b) Observer
2. Which participant(s) in (a) defined the `notify()` function? (1)
   Mediator
3. Which participant(s) in (b) defined the `notify()` function? (1)
   Subject
4. The - notify - function is defined in the Mediator participant in (a) and the - Subject - participant in (b). (2)
5. Write the code for the `change` function of the Mediator design pattern. (5)
   Write your answer to this question in the box provided in the next question.

QUESTION 6

5 points Save Answer

Write the code for the change function of the Mediator design pattern in this box.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

Paragraph Arial 10pt

P

0 WORDS POWERED BY TINY

QUESTION 7

15 points Save Answer

Create a UML Activity Diagram for processing an order.

Once an order has been finalised, four parties are involved in processing it: Online Sales, Accounting, Shipping, and Printing. Online Sales sends the order to Printing, where the associated PDF file is inspected. If the file is not suitable, a new file is requested from the customer. Once the file is suitable, Accounting is informed to charge the credit card. While this is done, Printing carries out the actual printing, and sends the result to Shipping. When Shipping has both received the printed products and confirmation from Accounting that payment was successful, the products are shipped to the customer.

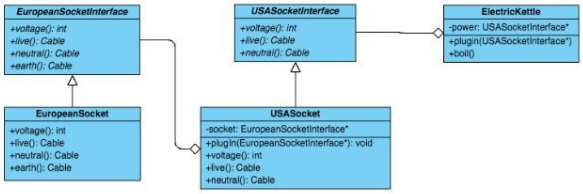
Upload your UML Activity diagram for the given scenario as a PDF.

Attach File Browse Local Files Browse Content Collection

QUESTION 8

6 points Save Answer

Consider the following UML Class diagram and answer the questions that follow.



1. Identify the design pattern. (1)

2. Identify the participants of the pattern by filling in the table below. (5)

Class	Role
Kettle	
USASocketInterface	
USASocket	
EuropeanSocket	

QUESTION 9

9 points Save Answer

You have a TV and its remote. The remote only has two buttons, one for toggling between on and off, and the other to change between the 5 programmed channels on the TV. A testing code snippet from the client perspective is given by:

```
TVRemote* tvr = new TVRemote;
tvr->onPushed();
tvr->channelChanged();
tvr->channelChanged();
tvr->channelChanged();
tvr->onPushed();
tvr->channelChanged();
tvr->channelChanged();
```

1. Which participants of the Command pattern do the TV and TV remote represent?

   TV -

   TV remote -

2. Given that the Command participant is defined as follows:

```
class Command {
    public:
        virtual void execute() = 0;
    protected:
        virtual char* getStatus() = 0;
};
```

   a. Suggest the names for the other participants in the *Command* hierarchy.

   b. Complete the code snippet below for the class definition and implementation of the functions of the TV remote.

b. Complete the code snippet below for the class definition and implementation of the functions of the TV remote.

```
class TVRemote {
    public:
        TVRemote(){
            button1 = ------(i)------;
            button2 = ------(ii)------;
        }

        void onPushed(){
            ---------(iii)----------
        }

        void channelChanged() {
            ------------(iv)-----------
        }
    private:
        -----(v)--- * button1;
        -----(v)--- * button2;
};
```

(i).

(ii).

(iii).

(iv).

(v).