

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2024

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BEng Honours Degree in Mathematics and Computer Science Part II
MEng Honours Degree in Mathematics and Computer Science Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

*This paper is also taken for the relevant examinations for the
Associateship of the City and Guilds of London Institute*

PAPER COMP50002

SOFTWARE ENGINEERING DESIGN

Monday 29th April 2024, 10:00

Duration: 120 minutes

Answer ALL TWO questions

Paper contains 2 questions
Calculators not required

1 Design Patterns

For this question, look at the code in the Q1 directory, related to streaming services.

You have a class `Subscription` which can be constructed with different options (for example, the number of users, the video stream quality, whether the subscription includes movies, comedy, sports, etc).

- a Write a new class `SubscriptionFactory` offering three options. It should allow creation of:
 - * Basic subscriptions, which just have news and comedy, in standard definition, and only allow a single device.
 - * Sports subscriptions, which have football and formula one content, in high definition, and allow two devices.
 - * Premium subscriptions, which have all types of content, in ultra high definition, and allow up to five devices.
- b Subscription services are renewed monthly, charging the user an appropriate fee. To model subscriptions effectively we will need to store a payment method when we create the subscription. We will allow both credit card payments and direct debit payments.

Use the Strategy pattern to allow our `Subscription` class to model subscriptions with different types of payment methods. Update the `renew()` method in `Subscription` to use your new payment method. Make your new payment methods print something to show when they are charged.

Update your `SubscriptionFactory` so that we have to provide the payment method when we create a subscription.

- c Your `Subscription` constructor might be getting quite large now, with lots of parameters. Apply the Builder pattern to make construction of `Subscription` objects neater.
- d Subscription renewals are charged according to the following formula:

A fixed cost of £2, plus £1 for each device allowed, plus an additional surcharge of £2 if the subscription includes football.

Write some units test that demonstrates that when we call the `renew()` method on a `Subscription`, the payment method is charged appropriately.

The four parts carry, respectively, 20%, 30%, 25% and 25% of the marks.

2 Web Services

For this question, look at the code provided to you in the directory Q2. This code implements a very simple web service for managing flight bookings.

If you open the class `SimpleWebService` and run the `main()` method, then you should be able to open a browser at <http://localhost:8888/bookings> and see some output.

You can vary the output by providing some parameters in the URL, for example:

<http://localhost:8888/bookings?customerId=123&startDate=01-05-2024&endDate=01-08-2024>

- a Look at the code as given, and...
 - i) Name the design pattern followed by `BookingsApiServlet`.
 - ii) Name the design pattern implemented by `BookingsDatabase`.
 - iii) Describe which level of the Richardson Maturity Model best characterises this service, and why.

- b Refactor the code of the web service to follow the Model-View-Controller architecture. Show the code for each component and briefly describe their role.

Assume that the `BookingsDatabase` class is owned by another team, and we are not allowed to edit its source code.

- c
 - i) Describe briefly what is meant by a “seam”
 - ii) Describe briefly what is meant by “sensing”
 - iii) Taking your main model class from part b), refactor the code to allow you to test this class in isolation. Write at least two tests.

Again, assume that we cannot edit the code of `BookingsDatabase`.

The three parts carry, respectively, 20%, 30% and 50% of the marks.