**Question 1** Answer the following questions about object patterns:

1. Describe in detail the singleton pattern; explain its uses and issues associated with thread safety and object construction? [10 Marks]

**The singleton pattern is a constructor object pattern which makes one and only one copy of a class and makes it available via a static interface. There are two approaches to object creation: at class load or on demand (lazy initialization). Lazy initialization has the benefit of allowing the constructor to take arguments. [4 Marks]**

**Possible applications of this are: a database connector/handler or some other resource handler where the resource should have only a single interface, so that requests can be shared out between clients. [2 Marks]**

**Singletons need to have thread safety built into their design, since the code can involve multiple threads sharing the same data. [1 Mark] One aspect of thread safety is the construction of the object itself, it is important that 1 and only 1 copy of the object is created and served to each of the calling threads. [1 Mark] One approach to this is to make the method creating the object synchronized, since this will ensure only 1 thread can call it at a time. However this is time consuming since synchronized methods are a lot slower (than methods without synchronization). Another approach is where the method uses a double locking approach, this involves only applying the lock as and when it is needed, when the object it being constructed for the 1st time.[1 Mark]**

1. Describe with the aid of an example, the Model View Controller Architecture.

[10 Marks]

**[TOTAL 20]**

**Model**

This stores application data as well as handle all the business logic, rules (e.g. who can access a student's transcript) for the system. It is also responsible for validation (can also be in controller), data persistence, application state (for example keeping a reference to a user’s shopping cart in an online session). **[2 Marks]**

**View**This part of the code renders the data to a format that can be presented to the user, for example for a web application the view code would generate HTML code that was sent to the user’s browser. **[2 Mark]**

**Controller** This part of the code interprets user input (such as mouse clicks or keyboard input) and sends it to the model. For GUI interfaces, each on screen widget capable of input, has typically an associated piece of controller code. **[2 Mark]**

**MVC web-mail example**

**[2 Marks for diagram]**

The user goes to the home URL for the mail service and the VIEW sends a user a login page in HTML to the user’s browser. The view code could render different versions of the HTML or use CSS to work with different sizes screens etc. The user types in there user/name and password and presses the submit button. The controller software intercepts the submitted Form page, constructs a Login command object and sends it to the Model command handling code. The Model checks the user’s credentials against the database and then makes a request to the view If the user’s credentials fail, the VIEW is requested to generate a failed login page. If the login is successful, the model code retrieves the user’s current email inbox from the database and makes a request to the View to display this data to the user. The VIEW generates the HTML which is sent back to the user’s browser. **[2 Marks for explanation]**

1. What benefits are gained by the use of the Model View Architecture? **[5 Marks]**

Makes it easier to change the user interface, (the VC) code without having the modify the model code. [1 Mark] This is because the VC code can be changed without exposing the critical business logic to change.[1 Mark] The MVC architecture makes it simpler to support multiple interfaces to the application. The module can present a standard public interface that the different VCs can be connected into.[2 Marks[ Using MVC the software can be developed by two separate teams, one’s with skills in GUI and others in skills in working with database technologies and the application area. **[1 Marks]**

**[TOTAL 25]**

1. An e-commerce website needs to be able to provide an interface to a number of different credit card payment systems. For the initial implementation there are 3 providers Metabank, Safepay and Paypoint each of which provides a different API . There needs to be a standard interface to charge for payments, code for standard validation of card details (example card number validation) as well as code to interface with the 3 APIs.

With the aid of a class diagram, show how the use of a wrapper/provider pattern can be used to solve this problem. Include in your answer how the interface and coding requirements are accommodated into the class structure. [10 Marks]



 **[6 Marks for diagram]**

Top level of design is interface which defines public methods to provide card service, example method given in this case is chargeCard this tries to charge money against a credit/debit card. [1 Mark]

The sub class below this is an abstract base class, this contains all the code

common to all the card providers, so it can contain for example methods to validate card details, example validateCardNumber() as well as implementing the public methods in the interface. [1 Mark]

The abstract base class also defines an abstract method called onChardCard, this is overridden by the clients of this class, so that they can provide service. [1 Mark]

The concrete implementations of the class do the requests to the card providers via the card providers API, this list can be added to without changing the external interface to the service. [1 Mark]

 **[TOTAL 35]**

**Question 2** At planning time a project is originally planned to take 40 staff months of development time and is split into equal length 4 phases each 1 month long with 10 staff on the project.

1. If task 1 runs ½ month over schedule, work out using 2 different assumptions how many more staff will be required to complete the project on time. [4 Marks]

The first task instead of using up 10 staff months uses up 15 staff months, since ½ x 10 = 5 man/months. Assuming only the first task is over schedule, we need to complete the remaining 30 staff months in 2½ months = 30/2.5 = 12 staff, so 2 extra staff will be required. **[2 Marks]**

If we assume the whole task has been underestimated, total man months left is now 30x1.5 = 45. (Assuming an increase of 50%). This means the staff needed to complete on time is now 45/2.5 = 18 staff, so 8 extra staff would be needed. **[2 Marks]**

1. Give TWO reasons why adding more staff often not lead to the project being delivered on time. **[2 Marks]**

**Reason 1 Time will be spent with the training the new staff getting them to know the software, this will take time from existing staff. [1 Mark]**

**The task by not be able to be split in such a way as to allow the new staff to be useful.**

**[1 Mark]**

1. What apart from adding more staff, discuss two alternative approaches when faced with an overrunning project. **[4 Marks]**

It is possible to re-schedule the project, allowing enough time to complete it to achieve its goals and with enough time for testing. However this will generally involve an increase in cost. [2 Marks] The other possibility is to trim the scope of the project, however this is only realistic if the core requirements of the project can be achieved. This can only be determined if the requirements themselves have been given appropriate priorities.

**[2 Marks]**

1. Explain what is meant by the term “anchoring” in the context of planning and development meetings. **[5 Marks]**

**Anchoring is where particular stake holders express particular views in how long development will take in an open meeting, this can lead to other stakeholder being persuaded to change their estimates based on this opinion. For example management might express a view that the development can be done in a very short time and others within the meeting might feel pressured to agree with this estimate. High anchors and estimates might be introduced by developers who are not aware of how the latest technologies could be applied to increase productivity. Some development planning techniques how to get away from estimation anchoring by letting everyone express their opinion at the same time (example poker planning). [5 Marks]**

**[TOTAL 50]**

**Question 3** In his paper "No Silver Bullet — Essence and Accidents of Software Engineering" Brookes named the following aspects of software engineering that make it difficult: complexity (accidental and essential), conformity, changeability and invisibility.

1. Explain what Brookes meant by the 4 aspects named above. [10 Marks]

Complexity of software is two-fold, the accidental complexity of implementing an idea in a programming language, in terms of all the coding instructions, data and interactions between the 2. The accidental complexity is a function of the programming language and tools used, for example an assembly language implementation may be more complex than a high level implementation due to the many coding instructions required.

Essential complexity of software engineering is the complexity of the problem itself, this is not effected by the programming environment and relates to the problem that requires solving, for example an voice recognition application poses a complex problem.

[4 Marks]

Conformity relates to the need for software to interface to other software or systems. This makes is hard to do, since there are many different external systems and standards which software may have to support, even if its own application is not very complex. [2 Marks]

Changebility Software is prone to constant changing requirements of the end user, this is due to many reasons not least being the difficulty of clearing determining requirements at the beginning of a project. [2 Marks]

Invisibility It is difficult to see clearly all the links and couplings in a software system, so module will call other module’s code but these calls are not visible at the called module site. This invisible complexity makes card hard to test, debug and modify. [2 Marks]

1. Explain how the use of high level programming languages and object patterns can be used to tackle the issues of complexity and conformity. [10 Marks]

High level programming languages reduce the instruction count making reducing the amount of code to read/understand and therefore complexity. **[2 Mark]** The code is written in a language which is closer to natural languages, this makes the problem solving easier as the language is designed to be closer to the problem domain.

**[2 Marks]**

Conformity is easier to implement via the use of high level languages since the high level languages allow you to hide the complexity within modules (encapsulation) as well as define standard interfaces to modules (interfaces). **[2 Marks]**

Object patterns define standard approaches to problems, this helps with complexity since for a given complex problem there may be an already suitable object pattern which will help solve it. **[2 Marks]**

Object patterns also help with conformity by allowing standard architecture to be developed for which different providers can be plugged into at a later stage.

 **[2 Marks]**

[TOTAL 70]

**Part B Answer 1 out of 2 questions, each question has 3 marks.**

**Question 4 a) Discuss the issues of lost update, deadlock and thread starvation in the**

**context of a concurrent system. [10 Marks]**

Proper description of lost update by multiple threads with simple example. [4 Marks]

Description of locking and deadlock with example. [4 Marks]

Description of problem of thread starvation with issue of the queuing of threads on monitors in Java as example. [2 Marks]

**b) Describe in detail the use of the Actor model and how it is can be applied to an application running over distributed system architecture.**

 **[20 Marks]**

**Clear description of actor model, including description of Actors, messaging, mailboxes and clear explanation of how this approach avoids the problems encountered with the shared memory model.**

**[10 Marks]**

**Description of types of mobility (weak and string) and how it is applied to actor concurrency. Application of mobility to the implementation of highly scalable architecture. [10 Marks]**

**Question 5 a) Explain the SCRUM development life cycle. [10 Marks]**

**SCRUM is an iterative development technique which splits the lifecycle into a series of development efforts called sprints. Each sprint lasts around 1 to 4 weeks and is preceded by a sprint planning meeting. [2 Marks]**

**At the start of the project and as the project moves forward, the total work needed to complete the project is contained within a document called the product backlog, as items are completed they are removed from the product backlog. This can include anything from product features to debugging tasks and technical tasks. Each item in the product backlog has a description/specification as well as a score in terms of its business value. Each item in the backlog is also given an estimate by the development team in terms of the effort required to complete it. [4 Marks]**

**In the sprint planning meeting it is decided the scope of work to be done, how long it is going to take, this work is added to the sprint backlog. The amount of work is carefully selected so that the it is enough to fill the sprint, this is done by looking at the workload delivered by previous sprints and what is called the projects velocity. Any work that is not complete at the end of a sprint is returned to the product backlog.**

**[2 Marks]**

**Daily SCRUM This is a short meeting in which all developers answer the following questions.. What have you done since yesterday? What are you planning to do today? Any impediments/stumbling blocks? [1 Mark]**

**As the project progresses its progress is measured using a burn down chart which gives you a clear indication of how much has been completed. [1 Mark]**

**b) Discuss and describe the benefits of the pair programming approach, in your answer refer to relevant research literature in the field.**

 **[10 Marks]**

**Two programmers work together to complete the programming task, one programmer writes the code while the other programmer, reviews the code, makes suggestions and comments on the approach. The two programmers swap places on a regular basis.**

**Pair programming provides the benefit of reducing risk, because the change of a design error or serious flaw passing 2 programmers working on the task is a lot lower than 1 programmer. A programmer working in a pair is a lot less likely to implement a “hack” or shortcut which is structurally unsound. The observer has the time to review the code while it is being produced, therefore ensuring higher levels of quality. The concept of collective code ownership is important here, it means that responsibility for code that is spread between developers. [5 Marks]**

**Looking at the research does not show conclusive benefits for pair programming, we can compare studies of Arisholm, Gallis and Sjøberg and Laurie Williams of the University of Utah.**

**The Williams et al. 2000 study showed that using pair programming there was a decrease in time to develop between 15% and 30% but this did require an increase in effort (programmer hours) between 15 to 60% with an increased in correctness of 15%.**

**Arisholm, Gallis and Sjøberg carried out research to determine how effective pair programming was in different contexts, for example with complex and relatively simple problems and with different combinations of staff skilling. They used a fixed set of problems and**

**split the developers into a pairing group and a single programming group.**

**They found that for most tasks the time taken was not significantly different when using a pair programming on average 84%, the amount of time was reduced but not by a large amount on average a reduction of 8%, however if the pairs were junior the increase in effort was much larger and the 111% and the time taken was larger as well.**

 **There was however a positive outcome in terms of correct solutions found overall of 7% and 73% for juniors.**

**One of the most interesting studies was a meta-analysis of many studies carried out by Jo E. Hannay et al. This showed small reductions in time (depending on the task complexity, less complex tasks have an improved time greater than complex tasks. They also showed that pair programming produces higher quality results for more complex problems. They also discovered a research bias in favour of pair-programming by some of the leading researchers in the area.**

 **In general pair-programming results are somewhat mixed it seems to be able to deliver code slightly faster, of higher quality but at considerable cost. However the overall cost of the product for its whole life-cycle should be taken into account so quality improvements earlier on in development could result in savings later on, these are hard to quantify with short term studies and a more longer term study may produce more conclusive results. [5 Marks]**

**c) Compare and contrast the use of XP user stories against classic UML user cases. [10 Marks]**

 **XP user stories are descriptions of functionality which has value to the end user. They are defined by short descriptions in text. In general user stories have to short enough so that they are easy to estimate and test and independent from one another so the development for them can be carried out separately.**

 **Use case descriptions have more structure and commonly defined in more detail than user stories. Within user cases there is a lot more added detail for example, user case exceptions describe what happens in exceptional conditions) as well as any pre-conditions before the case can happen and what will happen afterwards.**

 **In practise user stories are easier for end customers to get to grips with and provide an easy communication channel between the end user and development team. User case descriptions provide more detail to the developer. Without user case descriptions details such as how the UI works in detail could be lost. User cases also have the concept of inclusion which makes it easier to see re-use within the development. [10 Marks]**