Imagine a Hotel Management System, which would be a good example of a functional requirement?

1. *Each room should be described clearly on the hotel website.*
2. *A room can be reserved for a customer.*
3. *The system should have the latest anti-virus software installed.*
4. *The database should have enough storage for 10,000 customer records.*

Which of the following would be a functional requirement in a James Bond style car?

1. *Would be good to have the controls light up in fancy colours*
2. *The gadgets should have new software updates done every month*
3. *Must be able to spray oil out the back to stop any baddies chasing him*
4. *The stereo should be able to play extra loud*

In the generic software engineering process, domain modelling comes ....?

1. *before requirements.*
2. *After requirements.*
3. *after System design.*
4. *After testing.*

In an ATM System, which would be a non-functional requirement?

1. *Customer can select an amount to withdraw.*
2. *The interface should be physically robust and hard wearing*
3. *Cash must be ejected for the customer.*
4. *Customer can view their account balance.*

2. *Correct. This is basic functional requirement with nothing about how it should do it. The others have some description properties of the system or how functions should be performed.*

1

1. *Incorrect. This is a preference about how it should look, not an actual action it performs. The odd one out is Must be able to spray oil out the back to stop any baddies chasing him Because this states a function it should perform.*
2. *Incorrect. Updating the system is more of a preference and not a functional requirement The correct answer is Must be able to spray oil out the back to stop any baddies chasing him Because this states a function it should perform.*
3. *Correct. This states a function it should perform.*
4. *Incorrect. This is a preference about how the stereo should perform its function, not the actual function itself. The correct answer is Must be able to spray oil out the back to stop any baddies chasing him Because this states a function it should perform.*

2

2. *Correct. The order is: Requirements, Domain Modelling, System Modelling, Implementation...*

3

2. *Correct. This is a constraint on the performance of the interface, not an actual function the system performs.The others are simple, yet essential functional activities you'd expect from an ATM and so, are functional activities.*

4

Which of the following is the odd one out in a Book ordering System?

1. *A customer can browse available book titles*
2. *The layout of the menu can be toggled from a tabbed view to a drop down list.*
3. *A particular book name can be searched for*
4. *A customer can order a book*

At what stage of the generic software developed process does Requirements come?

1. *Last*
2. *in the middle.*
3. *Second, after Domain Modelling and before System Design*
4. *At the start*

How does Craig Larman describe use cases?

1. *Things the boss asks you to do*
2. *Things you say you're doing when the boss shows up*
3. *Things the boss does*
4. *Things you do when the boss is away*

In a spaceship control system, which of the following is the odd one out?

1. *The system should inform ground control its exact position, when requested*
2. *The system should provide a complete flight management solution.*
3. *The system should adjust pressure within the spaceship.*
4. *The system must have an autopilot mode to guide it to a given position.*

2. *Correct. This is just about how the menu options are displayed. No actual function being performed. The others are particular actions the system can perform.*

5

4. *Correct. Requirements are done at the start, generally speaking, however it is best practice for them to be done all the time, to adapt to any changes and correct any you may have wrong. But mostly done at the start.*

6

1. *Incorrect. This does not sound like too bad a description of what use cases are, but maybe boss should have to tell you what to do and so Larman describes them as Things you say you're doing when the boss shows up*
2. *Correct. Things you say you're doing when the boss shows up, so hes basically saying use cases are things the boss would expect you to be doing. Example: Boss: What are you doing? You: Printing accounts. Use case format: Printing (direct-verb) accounts (object-noun)*
3. *Incorrect. The boss doesn't do everything !! Larman describes them as Things you say you're doing when the boss shows up*
4. *Incorrect. You may well be doing worthwhile things while the boss is away, but Larman describes them as Things you say you're doing when the boss shows up*

7

2. *Correct. This is not at the same granularity as the others. it is very general and does not state any specific requirement. Where as the others are more fine grain, referring to exact requirements.*

8

In a James Bond style car system, which of these would not be a use case?

1. *Boost car speed*
2. *Eject passenger seat.*
3. *Spray oil out of car*
4. *Windows should be black, so no one can see in.*

In an Aircraft system which of the following is the odd one out?

1. *There must be a warning light when fuel is low*
2. *The warning lights for the pilot should be red and flash*
3. *Pilot should be able to disable a warning light*
4. *A warning light must indicate if the cabin door is not shut properly*

How can a domain class play a part in many use cases?

1. *It will always have the same role in them.*
2. *Each domain class represents the whole system, so by definition they do.*
3. *They always play a part in every use case. They just do.*
4. *It may have different roles in them or the same role.*

Which of these statements, best describes the requirements process?

1. *Requirements tell the software developer everything they need to know to develop the required system.*
2. *In the generic software development process, Requirements comes after domain modelling to aid the system class design.*
3. *Requirements are essential to deciding what software architecture to use.*
4. *Requirements are an essential input to the software development process.*

1. *Incorrect. This is a clear use case in the correct format. Boost(verb-direct) car speed(object-noun). The correct answer is Windows should be black, so no one can see in as this is more of a requirement of the car, not an action performed by the car in the format of verb-direct - object-noun, such as, Eject (doing, verb-direct) passenger seat (to something, object-noun).*
2. *Incorrect. This is a clear use case in the correct format. Eject (verb-direct) Passenger seat (object-noun). The correct answer is Windows should be black, so no one can see in as this is more of a requirement of the car, not an action performed by the car in the format of verb-direct - object-noun, such as, Eject (doing, verb-direct) passenger seat (to something, object-noun).*
3. *Incorrect. This is a clear use case in the correct format. spray(verb-direct) oil out of car(object-noun). The correct answer is Windows should be black, so no one can see in as this is more of a requirement of the car, not an action performed by the car in the format of verb-direct - object-noun, such as, Eject (doing, verb-direct) passenger seat (to something, object-noun).*
4. *Correct. This is more of a requirement of the car, not an action performed by the car in the format of verb-direct - object-noun, such as, Eject (doing, verb-direct) passenger seat (to something, object-noun).*

9

2. *Correct. This is a non-functional requirement, as it says how the function of the warning light should behave. The others are all functional requirements*

10

1. *Incorrect. They do not always have the same role in all use cases, they may have a different one or not play a role in some use cases at all.*
2. *Incorrect. Each domain class does not represent the whole system, they represent a domain within the system and so can play a role in many use cases, but they could only play a role in one.*
3. *Incorrect. They do not necessarily play the a part in every use case, they may only play a role in one.*
4. *Correct. They can play different roles in different use cases.*

11

1. *Incorrect. They play a huge part in what the the developer needs to know. However it is not everything. Activity diagrams, observations, interviews, documentation amongst other things can help aid this also. The answer is Requirements are an essential input to the software development process. Remember the hairdressing analogy where the hairdresser must know how the customer wants their hair cut. The haircut Requirements.*
2. *Incorrect. Domain modelling can not be done without any requirements gathering and so should be done before this. The answer is Requirements are an essential input to the software development process. Remember the hairdressing analogy where the hairdresser must know how the customer wants their hair cut. The haircut Requirements.*
3. *Incorrect. They could play some part in this decision, but by no means essential. The answer is Requirements are an essential input to the software development process. Remember the hairdressing analogy where the hairdresser must know how the customer wants their hair cut. The haircut Requirements.*
4. *Correct. Requirements come in at the start to help the development process begin. Remember the hairdressing analogy where the hairdresser must know how the customer wants their hair cut. The haircut Requirements.*

12

Which of these is not a very good use case?

1. *Review customer records.*
2. *Read customer records.*
3. *Customer records are reviewed.*
4. *Open customer records.*

Which of these would not be a good domain in a Parcel Delivery System?

1. *Parcel*
2. *Van*
3. *Customer Account*

Which of these would not be a domain class in an ATM Cash machine system?

1. *ATM*
2. *dispenser*
3. *keypad*
4. *screen*

Which of these is a good use case?

1. *Client Database*
2. *Add Client Record*
3. *Database update*
4. *Client update*

3. *Correct. This is not in a very good format, its noun (customer) then verb (reviewed) The others are of a better format as they say what we doing before the actual object.*

13

2. *Correct. A van might be used for the delivery, but wouldn't be used as a domain class. The others are well defined domain areas which the system will want to use.*

14

1. *Correct. This is the name of the system itself not domains within it. The others are domains which play their own individual parts in the ATM System.*

15

1. *Incorrect. This sounds like a domain object rather than a use case, its not doing anything, its just the name of something in the system. The answer is Add client record, which is a typical use case with the format of verb-direct - object-noun . Add (verb-direct) Client Record (object-noun).*
2. *Correct. This is a typical use case with the format of verb-direct - object-noun . Add (verb-direct) Client Record (object-noun).*
3. *Incorrect. It is not clear what this is, are we updating a database? And what database? Is it a database updating something else? If so what? The answer is Add client record, which is a typical use case with the format of verb-direct - object-noun . Add (verb-direct) Client Record (object-noun).*
4. *Incorrect. It is not clear what this is, are we updating a client? Is it a client updating something? If so what? The answer is Add client record, which is a typical use case with the format of verb-direct - object-noun . Add (verb-direct) Client Record (object-noun).*

16

Imagine an e-commerce system, which could be a functional requirement?

1. *Pictures of products on the website should be displayed in colour*
2. *Address labels should be printed on sticky paper.*
3. *Customer can add an item to their basket.*
4. *Customer Data should comply with the Data Protection Act.*

In a Bank System, which of these would not be a good domain class?

1. *Account*

3. *Tax System*
4. *Mortgage*

Requirements are ....?

1. *... not recorded at all. They are remembered from conversations.*
2. *... gathered at the end of the software process.*
3. *... recorded in an official document.*
4. *... a waste of time.*

Which of these is not a good domain class for a game system?

2. *Players Score*
3. *Box game comes in.*

1. *Incorrect. it is a preference on how pictures of products should be displayed. Remember the car analogy, the car colour made no effect what so ever of the its functional requirements. The correct answer is Customer can add an item to their basket as this is a sensible basic functionality you could expect of an e-commerce system.*
2. *Incorrect. If it was just Must be able to print Address labels It could be seen as a functional requirement However, it is a condition on this, that the labels should be on sticky paper. The correct answer is Customer can add an item to their basket as this is a sensible basic functionality you could expect of an e-commerce system.*
3. *Correct. This is a sensible basic functionality you could expect of an e-commerce system. The others are all some kind of constraints on how the business performs tasks and acts.*
4. *Incorrect. This is a condition on how customer data should be treated. not a function itself. The correct Answer is Customer can add an item to their basket as this is a sensible basic functionality you could expect of an e-commerce system.*

17

3. *Correct. Tax system would be an external system, a separate system. The others are common domains within a typical bank.*
4. *Incorrect. The correct answer is Tax system, as this would be an external system, a separate system. The others are common domains within a typical bank.Incorrect. The correct answer is Tax system, as this would be an external system, a separate system. The others are common domains within a typical bank.*

18

3. *Correct. Requirements are recorded in a semi-structured document. The others not serious answers at all*

19

3. *Correct. This is not important in the game system. The others sound like things used in a game.*

20

In a Banking System, which of these would you consider a non-functional requirement?

1. *Cash withdraws should be possible 24/7*
2. *Customer can withdraw Cash*
3. *Customer can transfer money to another account*
4. *Customer can close their account*

Which of these requirements for a hotel booking system is the odd one out?

1. *A customer can book a room up to 12 months in advance.*
2. *A customer can amend a booking.*
3. *A customer can cancel a room booking.*
4. *A customer should be able to manage their bookings*

A Domain Class can be used to ... ?

1. *Realise several use cases.*
2. *Realise only one use case*
3. *discover use cases*
4. *aid requirements gathering*

In a spaceship system, which use case would be the odd one out?

1. *Spaceship takes off.*
2. *Spaceman increases speed.*
3. *Spaceman creates an error log*
4. *Ground control reads spaceship location*

1. *Correct. This is how something should be done, not what should be done.*

21

4. *Correct. The level of abstraction is higher than the others. It is very general and can include many other things. The others are of a more fine granularity and refer to more specific requirements*

22

1. *Correct. One Domain class can play a role in many different use cases. For example in an ATM example. Imagine a domain class called display, you would expect this to play a role in achieving the following use cases: Display Balance and Withdraw Cash (to display amounts to withdraw)*
2. *Incorrect. A Domain class can help realise several use cases, not just one.For example in an ATM example. Imagine a domain class called display, you would expect this to play a role in achieving the following use cases: Display Balance and Withdraw Cash (to display amounts to withdraw)*
3. *Incorrect. Use cases will already have been discovered before, from requirements gathering, amongst other techniques*
4. *Incorrect. Requirements gathering would be done a while before domain classes are used. Remember the order of the phases: - Requirements - Use cases - domain modelling (which involved domain classes) The answer is A Domain class can be used to realise several use cases.*

23

1. *Correct. this is a very general function, which is likely to involve many other use cases, such as open rocket thrusters for example. The others are specific use cases.*

24

In a stock ordering system which of these is the odd one out?

1. *The system must be able to order a given stock item.*
2. *The system should help the company manage its stock levels.*
3. *Stock items should be displayed with a picture.*
4. *The system should be able to display stock levels.*

In the hairdressing analogy, which would be a non-functional requirement?

1. *hair must be cut short round the back*
2. *Don't cut too much off the top.*
3. *The chair height should set high*
4. *hair must be cut short round the sides*

In a Car hire system which of these would be a good domain class?

1. *Add Customer*
2. *Purchase new car*
3. *Car hire service*
4. *Car*

In a flight booking database system, what would be a non-functional requirement?

1. *A particular booking information can be read from the database*
2. *A new booking should be added to the database*
3. *The booking database should be backed up every 5 minutes*
4. *A cancelled booking can be deleted from the database*

2. *Correct. This is a very general requirement, where as the rest are more fine grain requirements.*

25

3. *Correct. This does not effect the function of the hair being cut. The others are specific to the function of cutting hair*

26

4. *Correct. A car would be an important domain area of the system. The others all look like use cases, as they are doing something to things.*

27

3. *Correct. this is more of a safety requirement and not a actual functional requirement. The others are actual functions on the database*

28

Which one of these requirements for a car hire company is the odd one out?

1. *Clients should be able to browse the vehicles available to hire*
2. *The system should allow clients with a way to hire cars*
3. *Clients should be able to view vehicle availability*
4. *Clients should be able to extend the time period of a vehicle hire, availability permitting.*

Why is DATABASE not a good domain class name?

1. *Its external to the system.*
2. *It does not say what the database is holding, CUSTOMER DATABASE would be better.*
3. *Databases can not be used as domain classes.*
4. *Its in capitals.*

In a university system, which of these would not be a good domain class?

1. *Student*
2. *Select Course*
3. *Exam*
4. *Course*

Which of the following would be a sensible use case in a banking system?

1. *Customers should receive an account statement every month*
2. *Customer withdraws money.*
3. *Customer manages account.*
4. *Customer records should be stored in a back up database.*

2. *Correct. This one is too general and not of the same granularity as the others.*

29

1. *Incorrect. It could be a big part of the system, but the problem with this one is it needs more description, such as CUSTOMER DATABASE, to give it more meaning.*
2. *Correct. Its more description, on its own, it means nothing.*
3. *Incorrect. Databases can be very good domain classes, but the problem with this one is it needs more description, such as CUSTOMER DATABASE, to give it more meaning.*
4. *Incorrect. It has nothing to do with it being in capitals. The problem with this one is it needs more description, such as CUSTOMER DATABASE, to give it more meaning.*

30

2. *Correct. This sounds like a use case. The others are typical domain areas of a university.*

31

1. *Incorrect. This seems like more of a requirements, a use case equivalent would be Send account statement, which follows the use case format of verb-direct object noun. Send (verb-direct) account statement (noun). The answer is Customer withdraws money. In this case it is Customer (Actor) withdraws (verb-direct) money (object-noun)*
2. *Correct. This is a clear use case in the format of verb-direct object noun - Withdraws Money, with the Customer being the actor performing this.*
3. *Incorrect. This is a very general use case and would comprised of many smaller, more fine grain use cases, such as Transfer Money, View account and withdraw money. The answer is Customer withdraws money. In this case it is Customer (Actor) withdraws (verb-direct) money (object-noun)*
4. *Incorrect. This seems like more of a requirements, a use case equivalent would be Back up customer records, which follows the use case format of verb-direct object noun. Back up (verb-direct) Customer records (noun). The answer is Customer withdraws money. In this case it is Customer (Actor) withdraws (verb-direct) money (object-noun)*

32

Which is the odd one out here?

1. *Staff member adds a booking.*
2. *Staff member deletes a booking.*
3. *Staff member manages hotel bookings.*
4. *Staff member changes a booking.*

Use Case: Tax system extracts tax payments from company database which is the actor in this company system?

1. *Tax System.*
2. *Company manager.*
3. *Company database.*
4. *Employee*

Which is the odd one out from these requirements for a music download website system?

1. *Members can download bought songs.*
2. *Members can leave comments on artists page.*
3. *Members can manage their account.*
4. *Artists can add songs to their catalogue.*

Imagine an e-commerce system, which would be a non-functional requirement?

1. *Customer can change their payment details*
2. *Orders can be cancelled*
3. *Orders should be confirmed via email within 24 hours.*
4. *Customer can view their order status.*

3. *Correct. This is too general. The others are of more fine granularity and refer to a specific task within the system.*

1. *Correct. The tax system is an external system that uses the company system.*

3. *Correct. This is not at the same abstraction level as the others. It is of a more general nature about what the whole system should achieve, rather than the more specific functionality of the others.*

3. *Correct. The time constraint on this makes it non-functional. The others are clear functions, with no constraints on how they are done. Remember the file download example, the constraints on speed of download, made no effect on the file being downloaded or uploaded.*

In A Banking System, which of these would you consider a functional requirement?

1. *Customer can create a new account*
2. *The system should have the latest security protection*
3. *Customer can view account changes within 1 hour of transactions happening*
4. *Customer can make balance transfers online*

What is a behavioural model?

What is a structural model?

What is an interaction diagram? Give an example of one.

1. *Correct. This is a clear functional aspect of a banking system.*
2. *Incorrect. This is more about the system performance and how it performs, rather than an actual function it does. The correct answer is Customer can create a new account as this is a a clear functional aspect of a banking system.*
3. *Incorrect. This is more about the performance of a function, rather than the actual function. If it said Customer can view account changes only, then that would be a functional requirement. The correct answer is Customer can create a new account as this is a a clear functional aspect of a banking system.*
4. *Incorrect. This is non-functional as it has the constraint of being online, rather than just the function itself. The correct answer is Customer can create a new account as this is a a clear functional aspect of a banking system.*

37

A behavioural model describes what must happen in the system. They are often used to ***describe the functionality of the system****. Include activity diagrams, interaction diagrams and use case diagrams.*

38

Describes what components are present in the system being modelled, and are therefore structural models are used when documenting the architecture of the system.

39

A type of behavioural model, it emphasises the flow of data through the system. A sequence diagram is a type of interaction diagram.

40

What is a system class diagram?

41

What is a domain class diagram?

42

What is a domain class?

43

What is the difference between a domain model and a system model?

44

A diagram that describes the classes in a system, their attributes, methods and the relationships between them. A type of structural diagram.

41

A diagram that describes the classes in a system, their attributes and the relationships between them. Unlike a system class diagram, it doesn't have methods. It's a type of structural diagram.

42

Each box in a domain/system class diagram is a domain class. It is a self contained component in the system, with its own name, attributes, methods and relations to other domain classes.

43

A domain model maps to real life entities (classes), while a system model maps to actual programmable classes. You don't have user interfaces in the domain model, but you do in the system model.

44

What does a sequence diagram show?

45

What is an activity diagram?

46

What is an action?

47

What is an activity?

48

How classes interact with each other and in what order these interactions occur. It is a type of interaction diagram. System classes go along the top, and interactions go horizontally bewteen vertical lines below them.

45

An activity diagram decribes what the system should do, not how it does it. Basically a flowchart of what the system should do.

46

A non-decomposable piece of behaviuour.

47

A composed set of actions. We only use activities in activity diagrams.

48

What is a non-functional requirement?

49

What is a functional requirement?

50

What is an actor?

51

What is a use case?

52

A creteria that is used to judge the operation of a system rather than specific behaviours.

49

A specific behaviour or function that the system should implement. Often described as a set of inputs, the behaviour and outputs.

50

An actor specifies a role played by a user or any other system that interacts with this system. Actors interact with use cases.

51

A list of steps, defining interactions between an actor and the system to achieve a goal. Use cases can have multiple actors, but always have a main actor.

52

What does WIMPS stand for?

53

What are the different styles of user interaction?

54

Windows, Icons, Menues, Pulldowns, Selection

53

Wizard interface
Form fill in
Direct manipulation (e.g. word, photoshop)
Command line
Batch processing (e.g. cron)

54