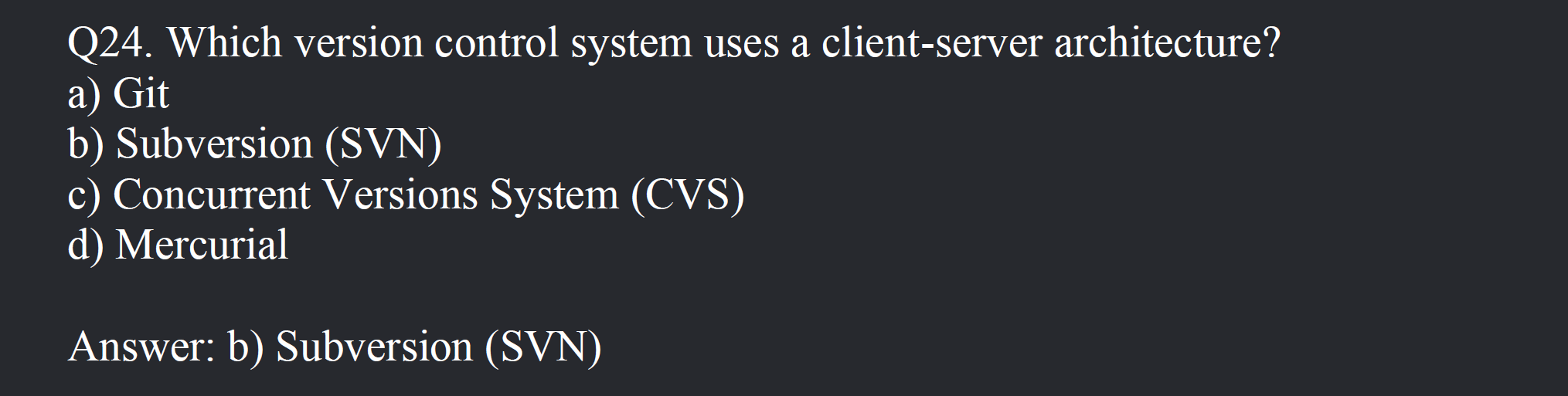
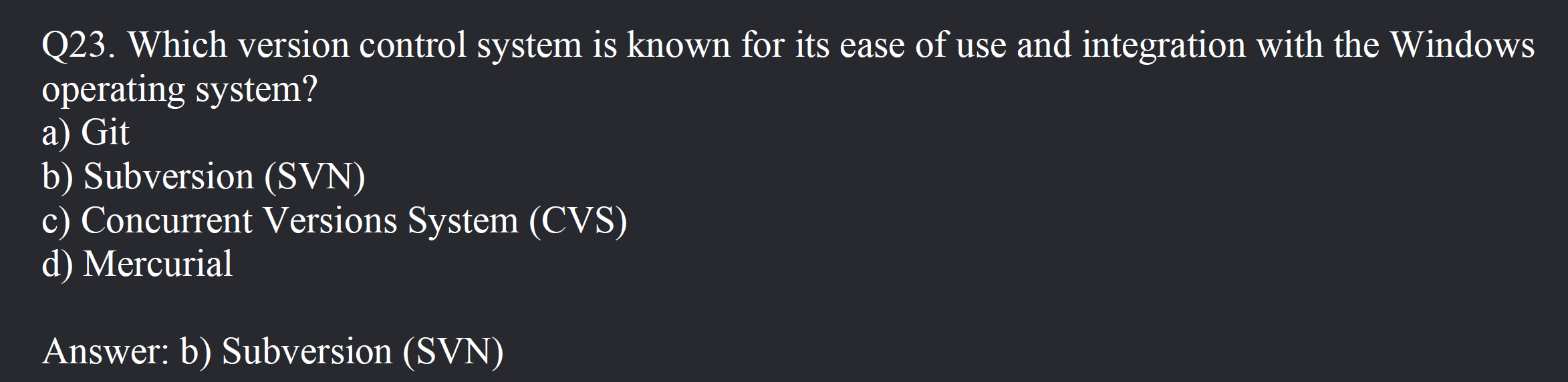
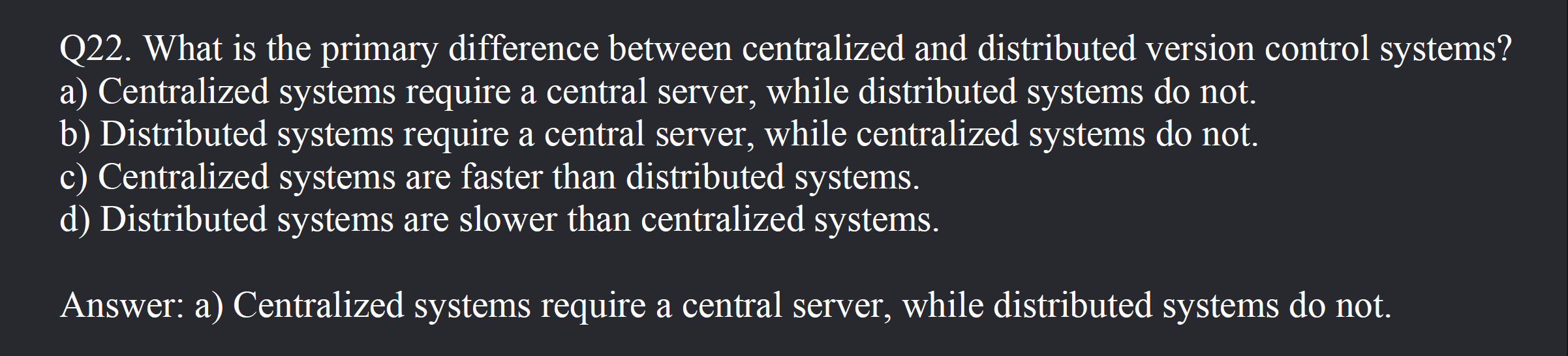
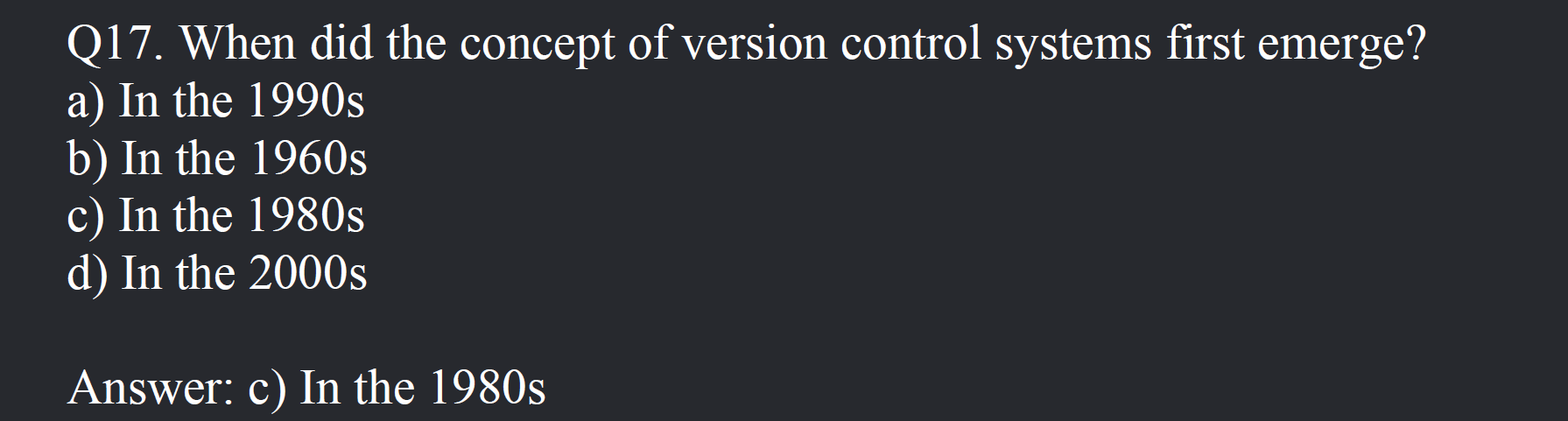
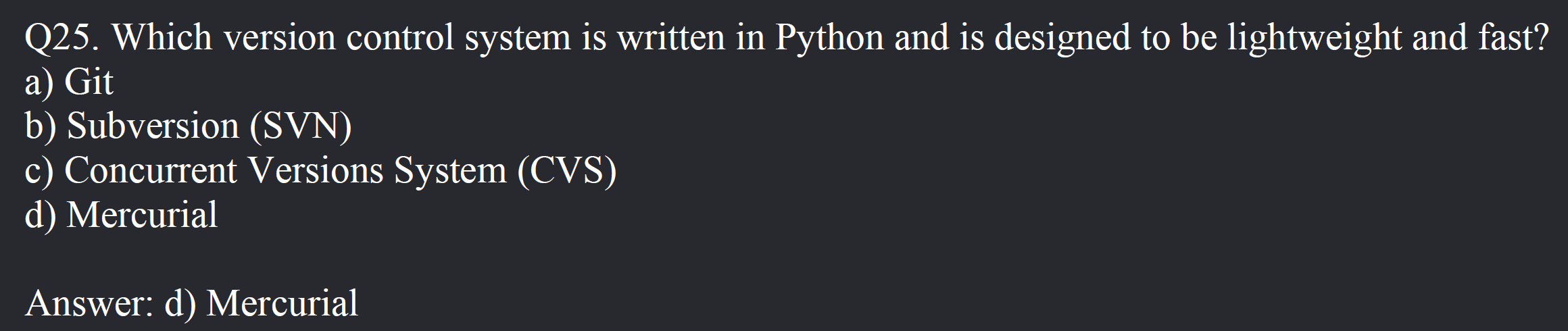
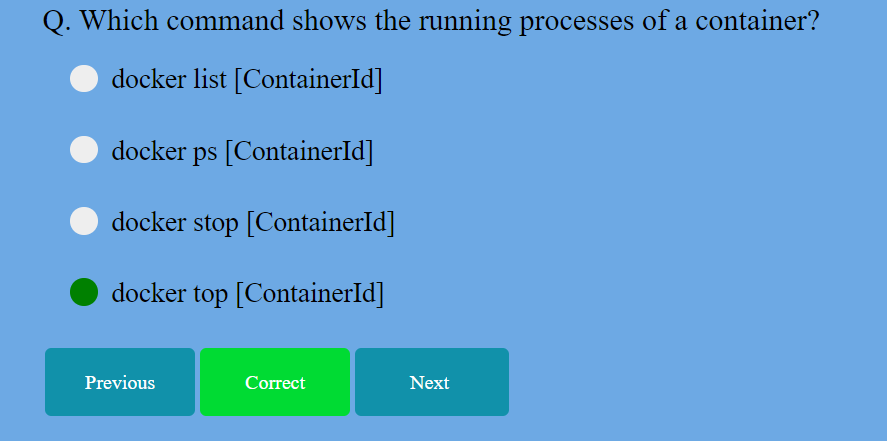
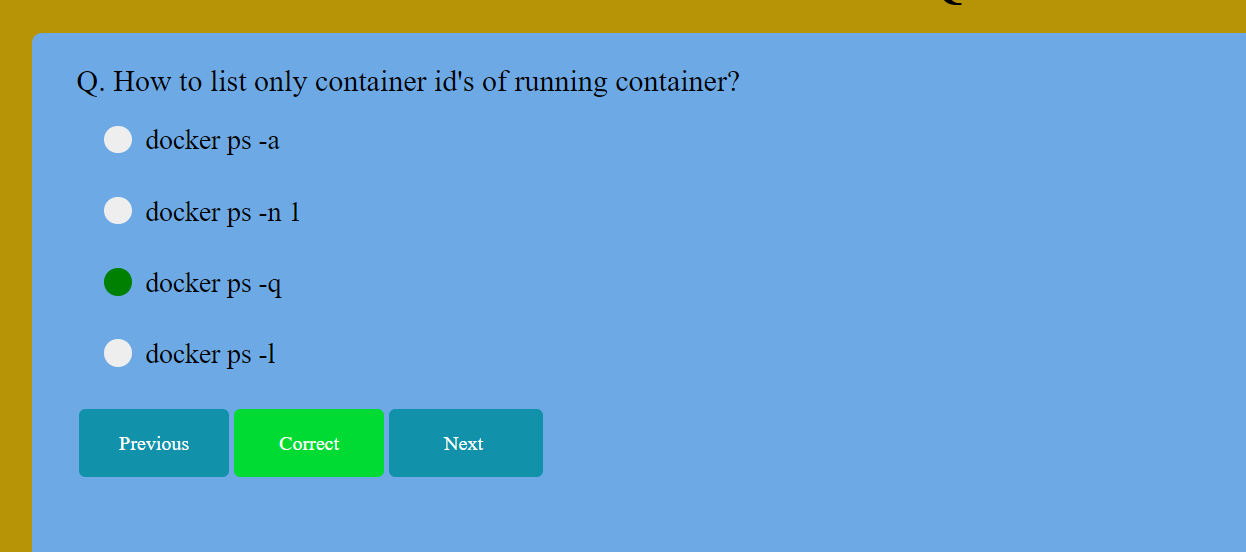
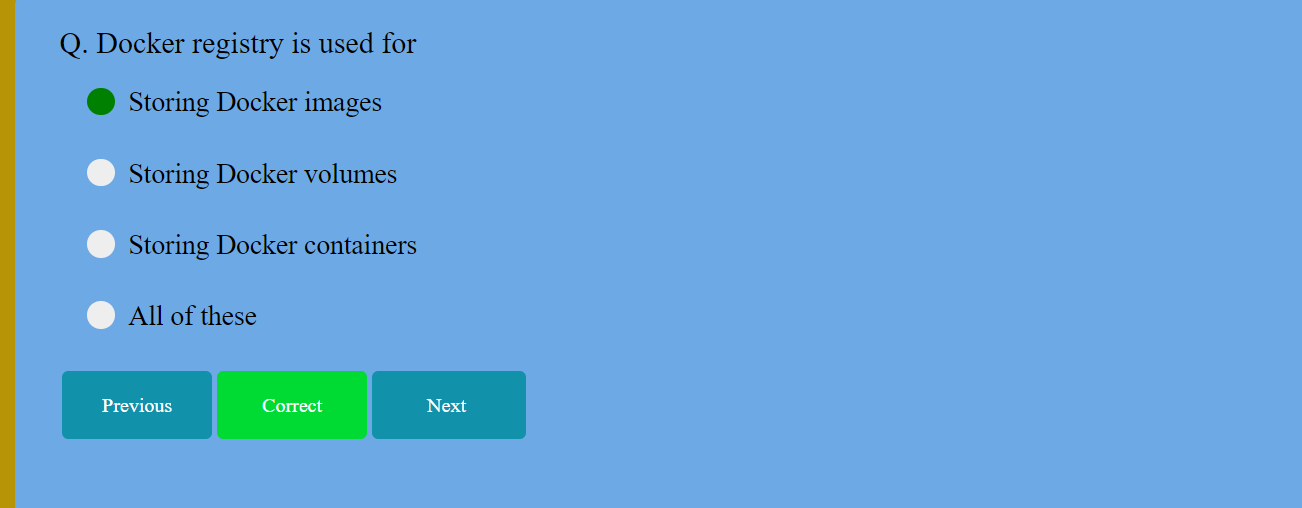
**SDM**

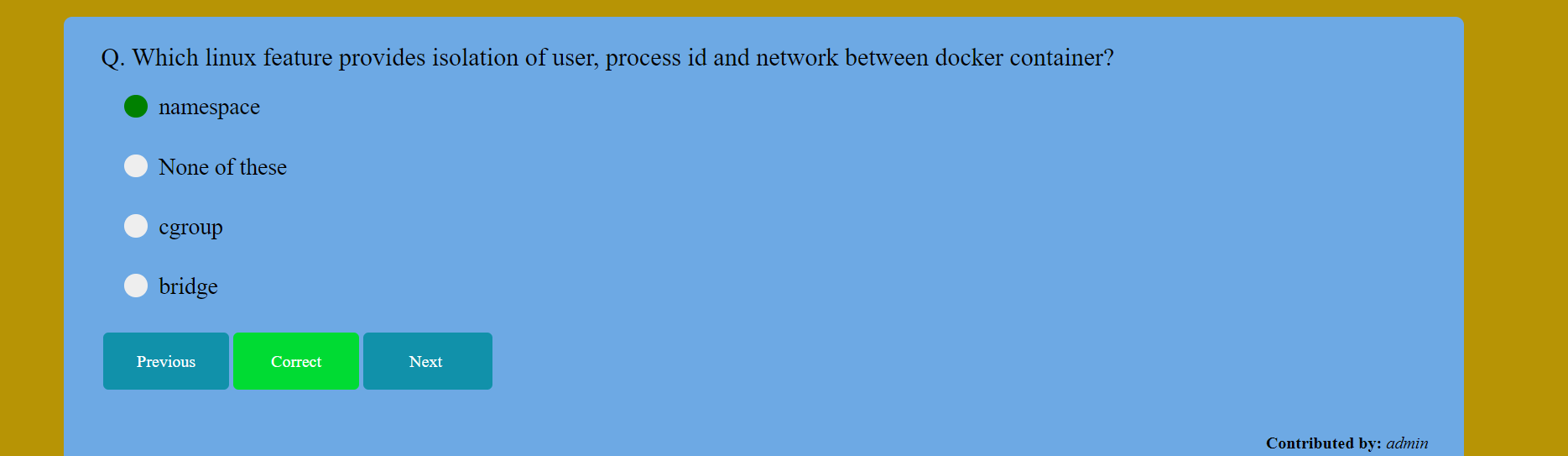


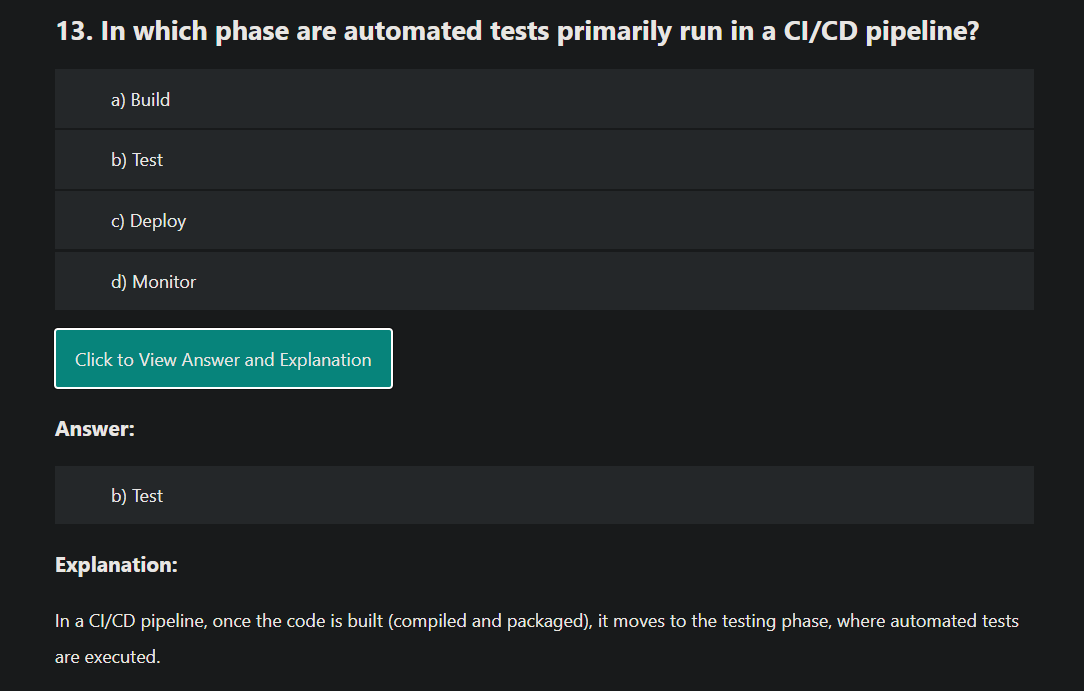


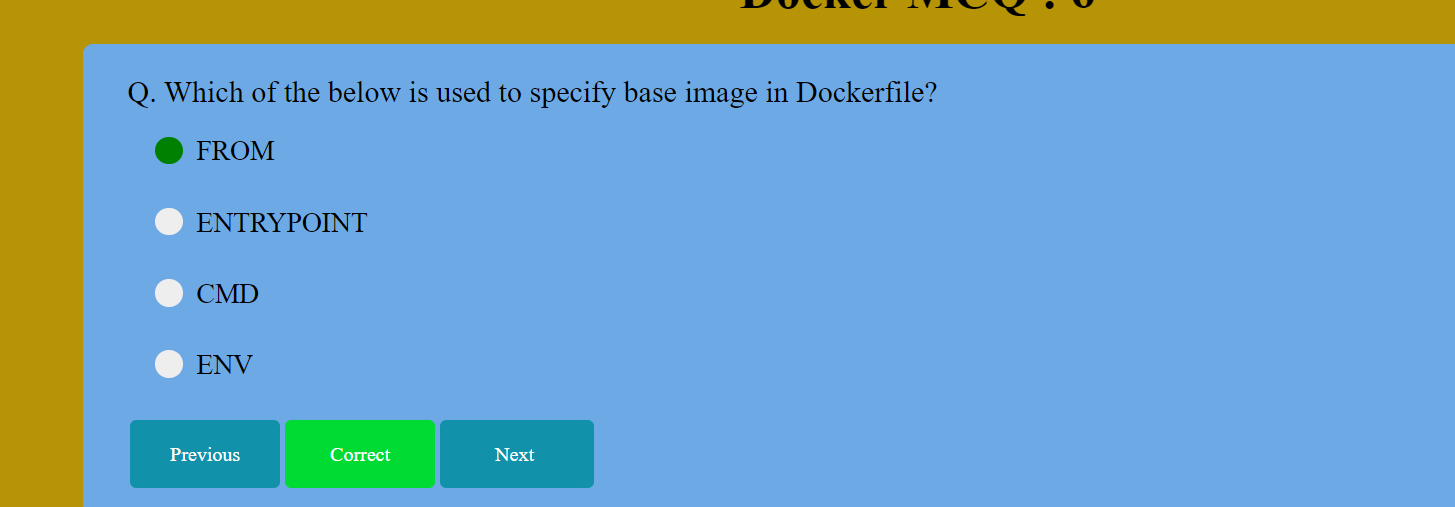
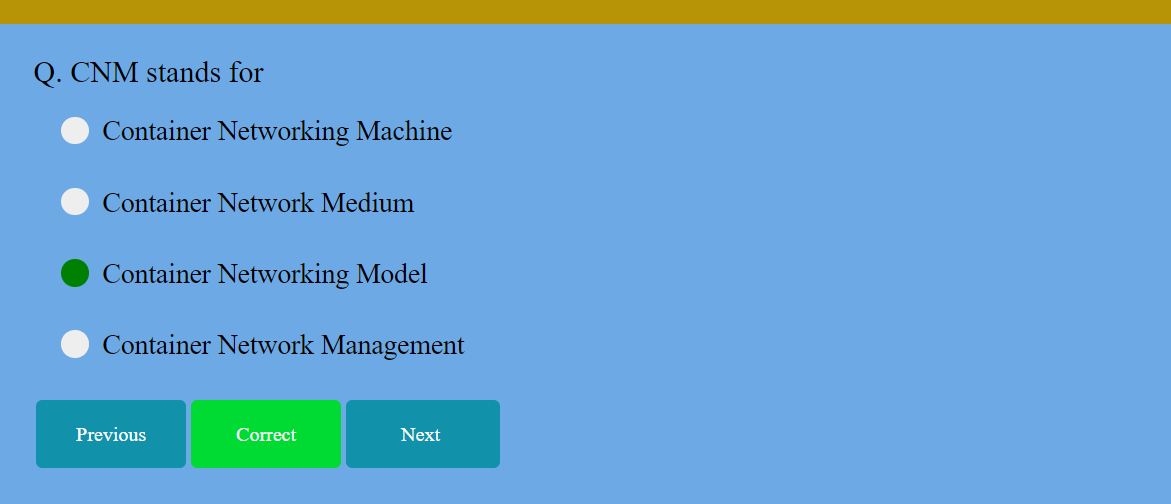


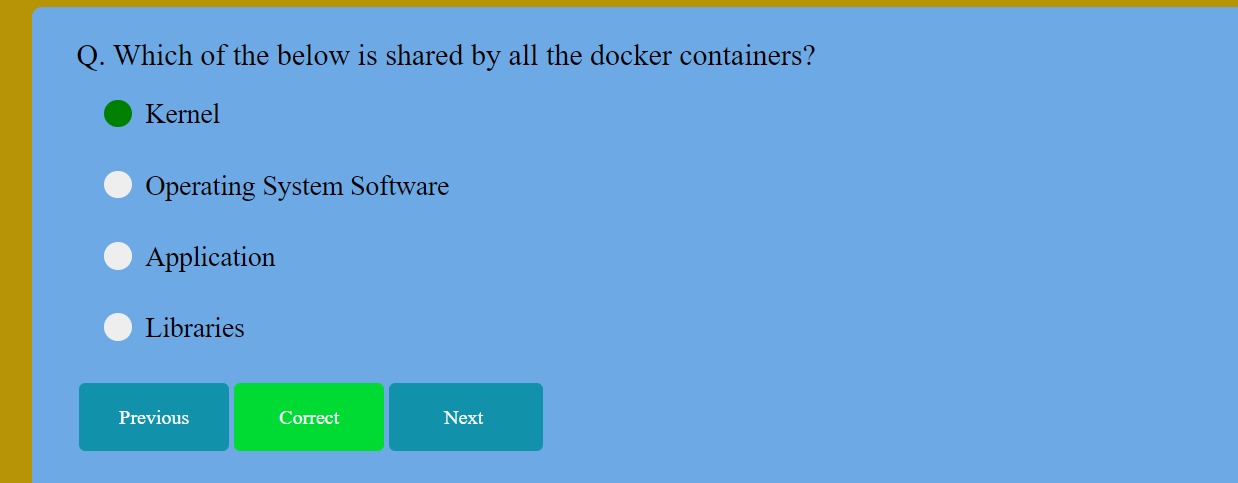


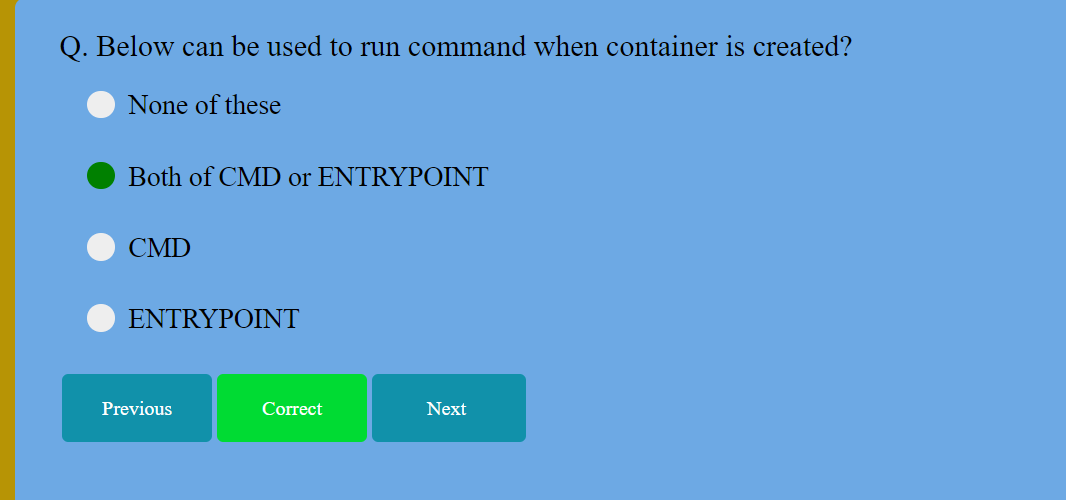


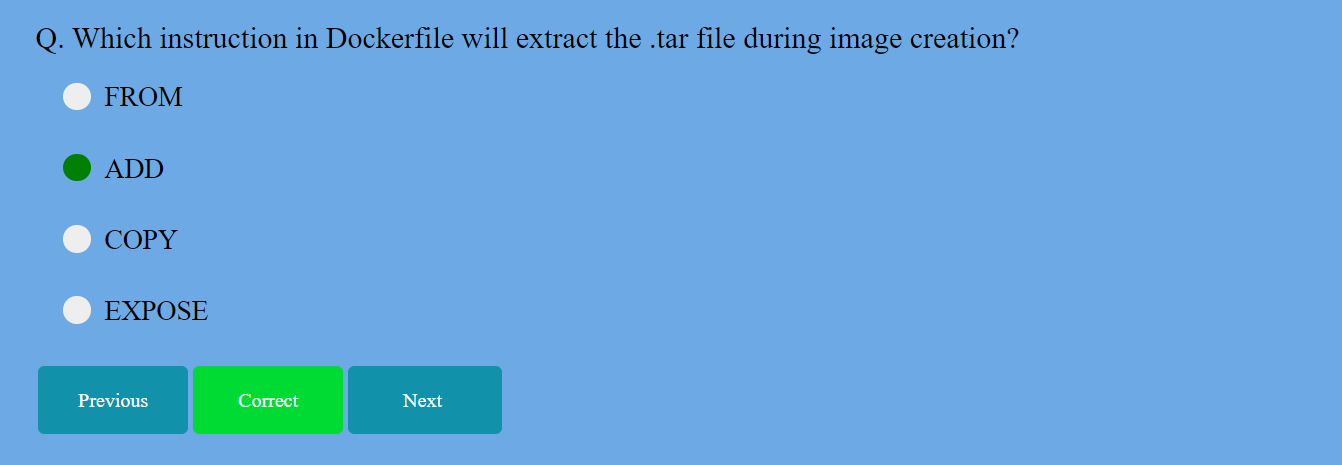


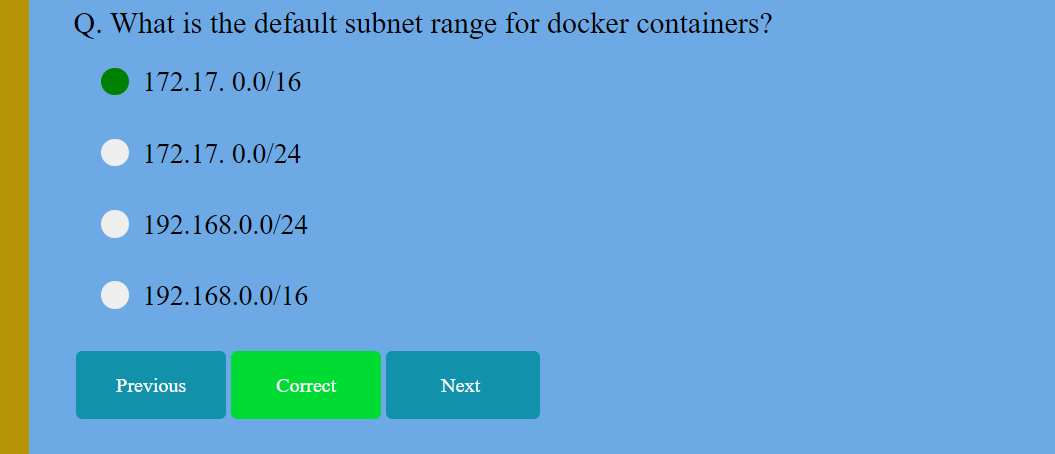


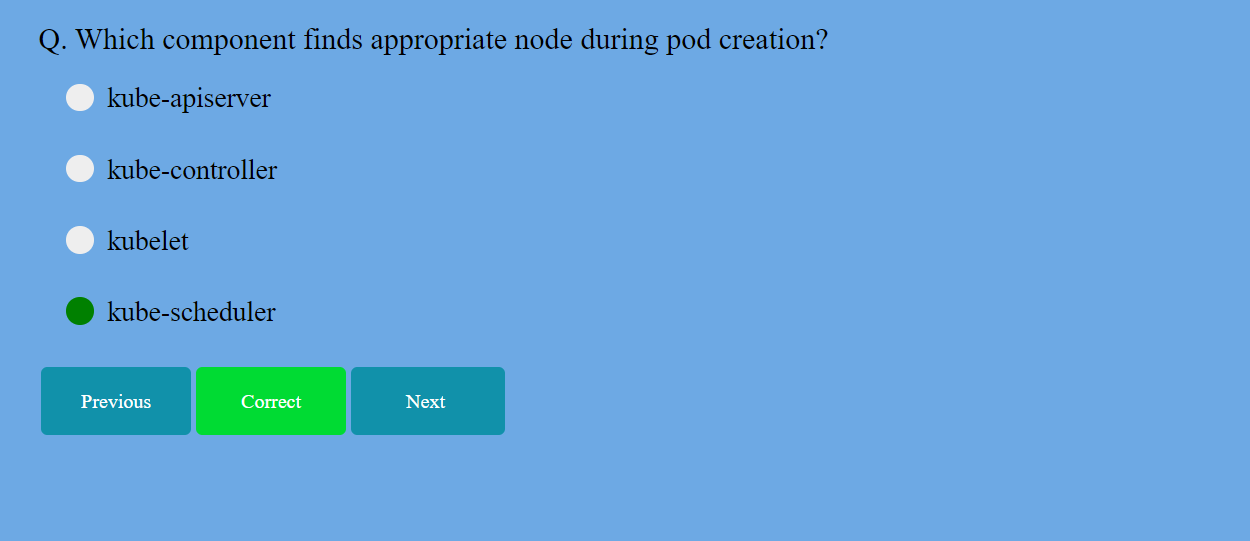


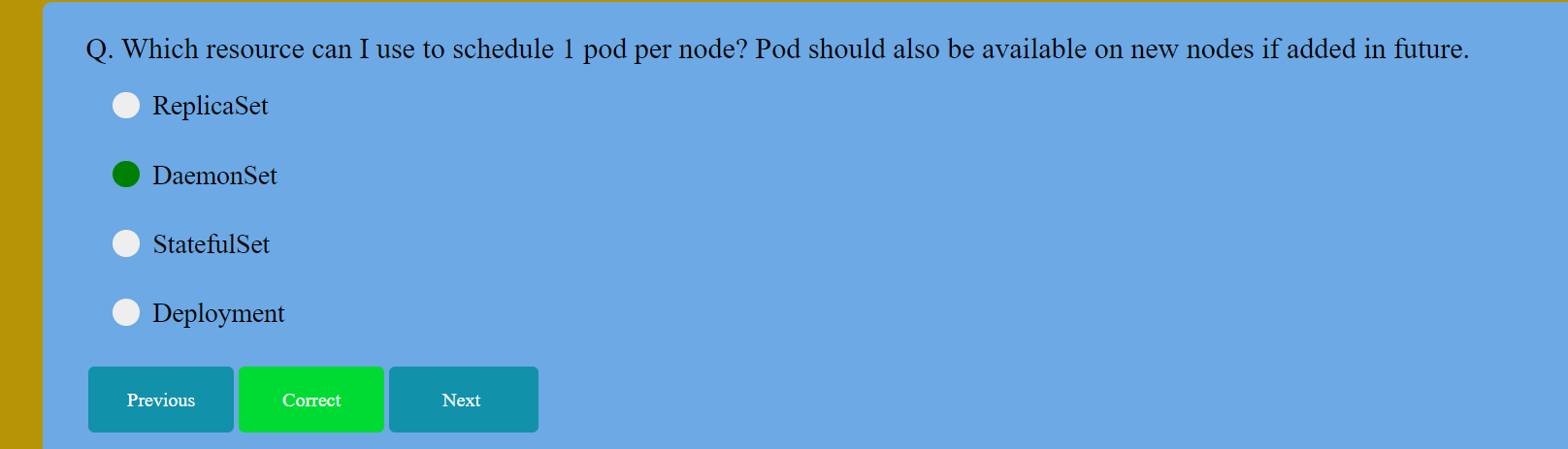


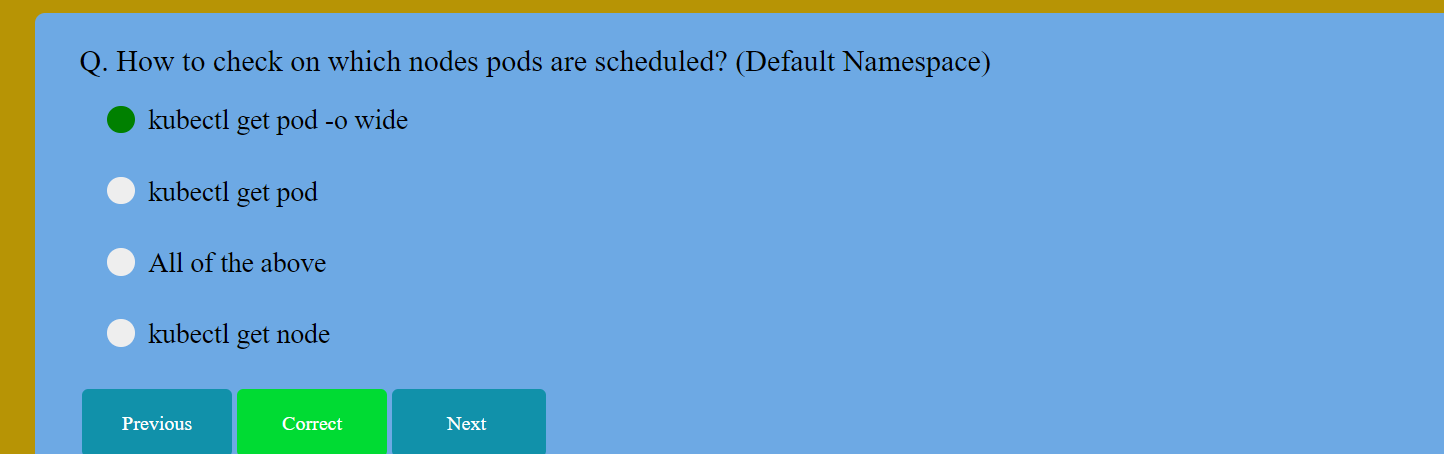


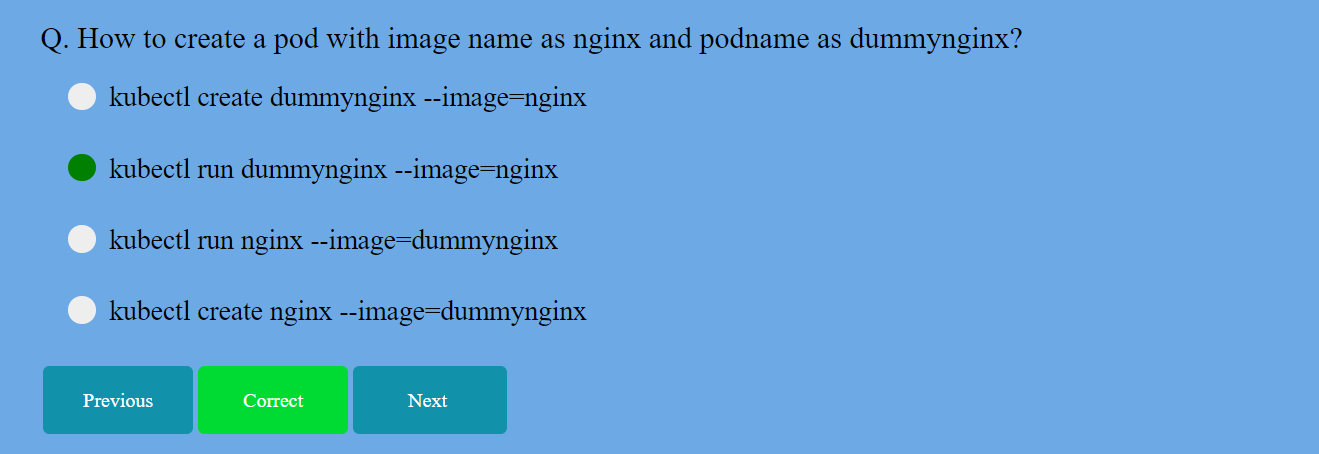


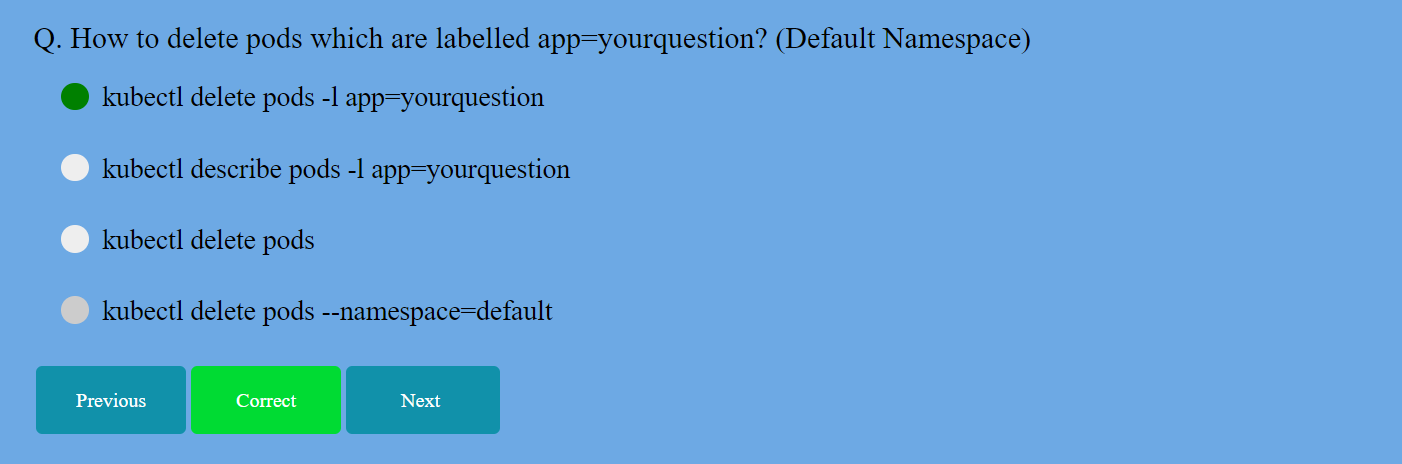






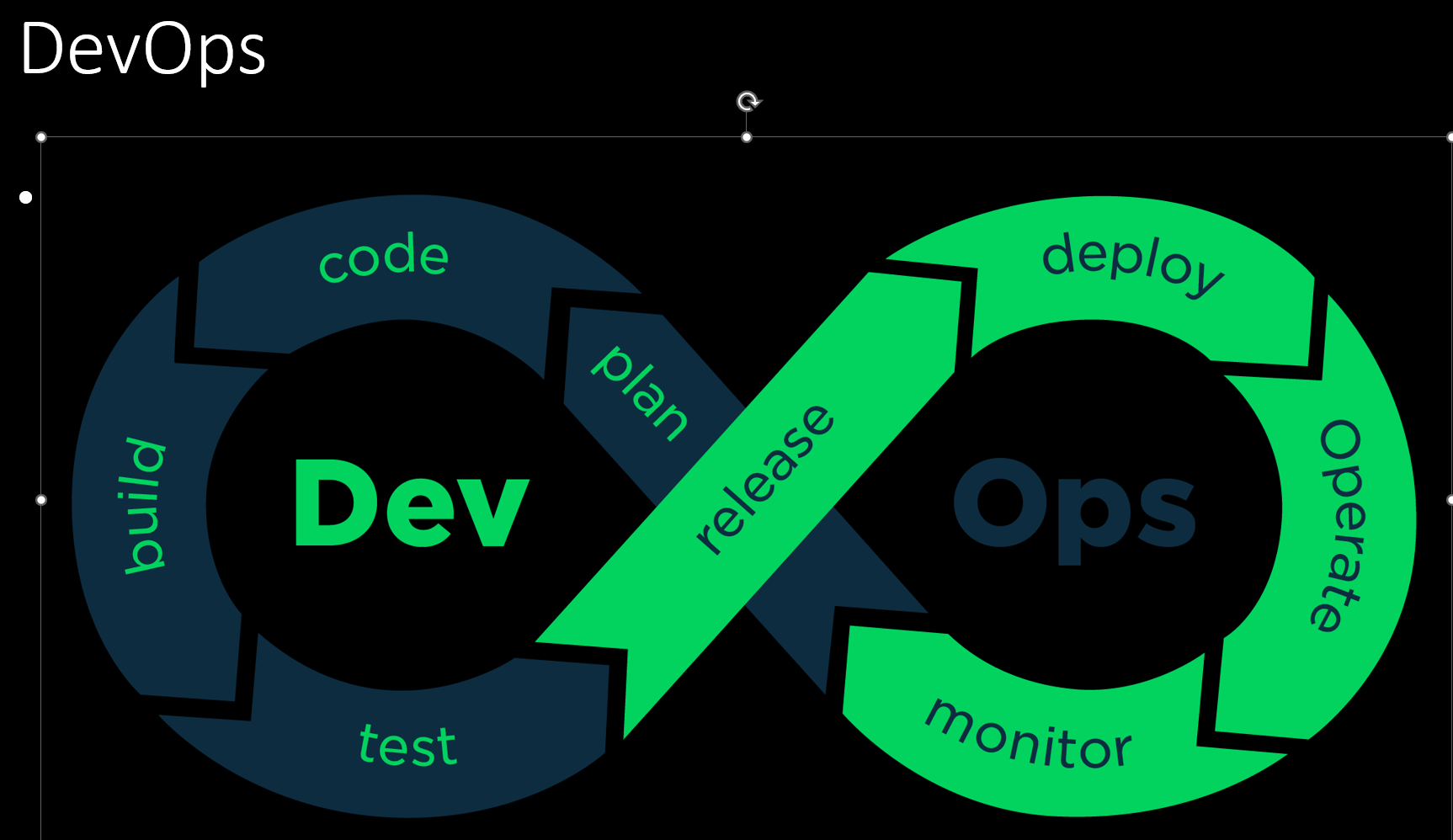






SDM MCQ Bank first 25 done till here

Sir ppt started devops



Stateful vs stateless: a comparison

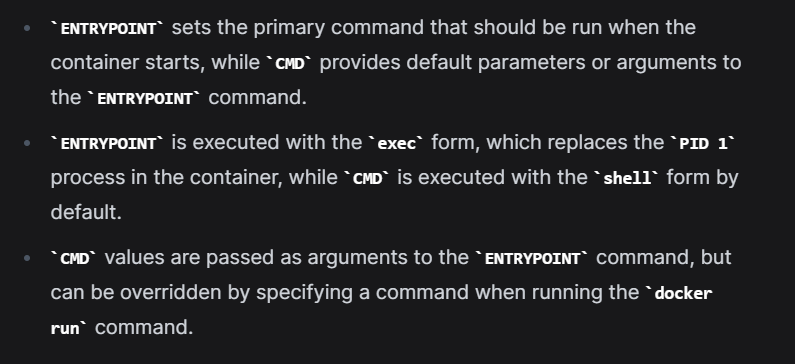
The key difference between stateful and stateless is whether an application retains information about the current state of a user's interactions or if it treats each request as an independent, isolated transaction. However there are also specific differences including:

Scalability: Stateless applications are generally more scalable, as each request is independent and can be handled by any available server. Stateful applications may require more complex mechanisms for load balancing and session management.

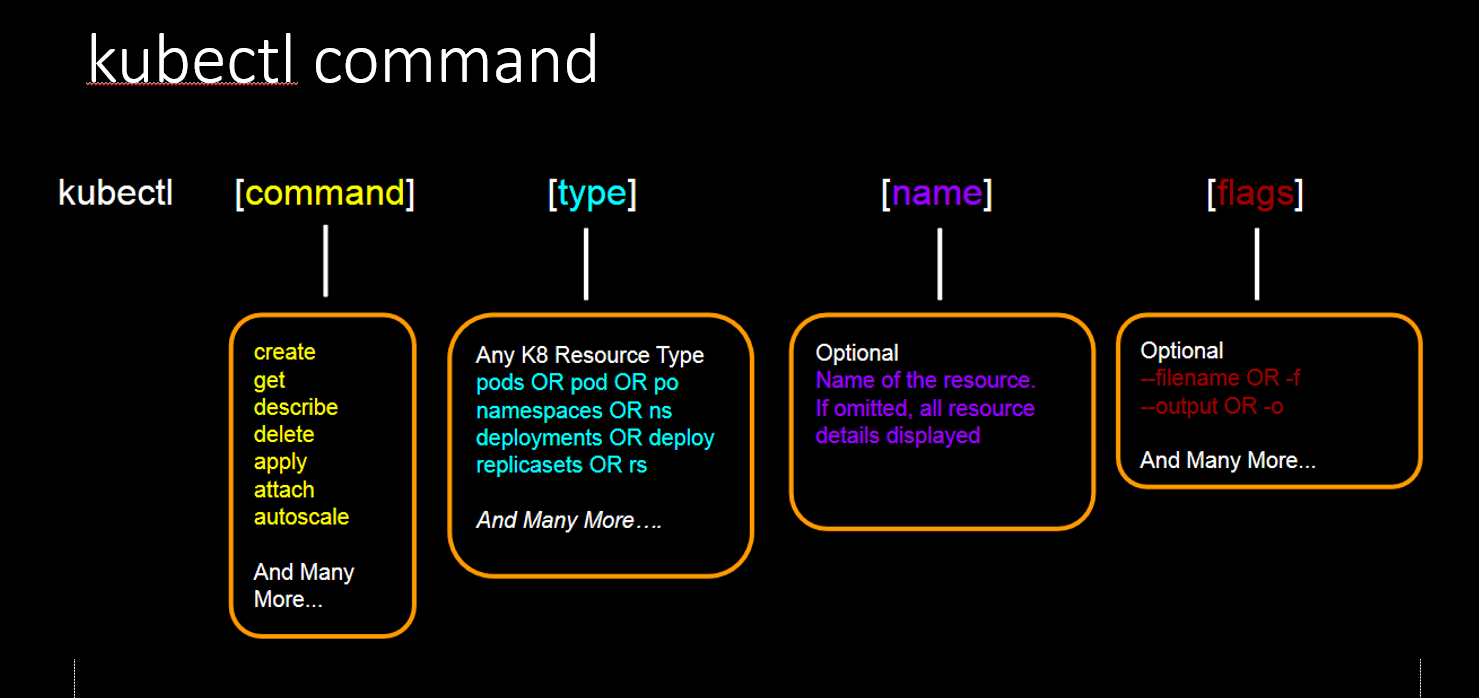
Fault tolerance: Stateless applications can be more fault-tolerant, as the loss of a server doesn't impact user sessions. In stateful applications, the loss of a server can result in the loss of session data unless additional measures, such as session replication or clustering, are in place.

Resource utilization: Stateless applications often have lower resource utilization because there is no need to store and manage session data. Stateful applications may require more memory and processing power to handle and maintain session information.

Development complexity: Stateless applications can be simpler to develop and maintain, as there is no need to manage state across multiple requests. Stateful applications, on the other hand, require careful handling of session data and state management.



Kubectl 🡺 cmd line utility which helps to communicate with clusture



Git

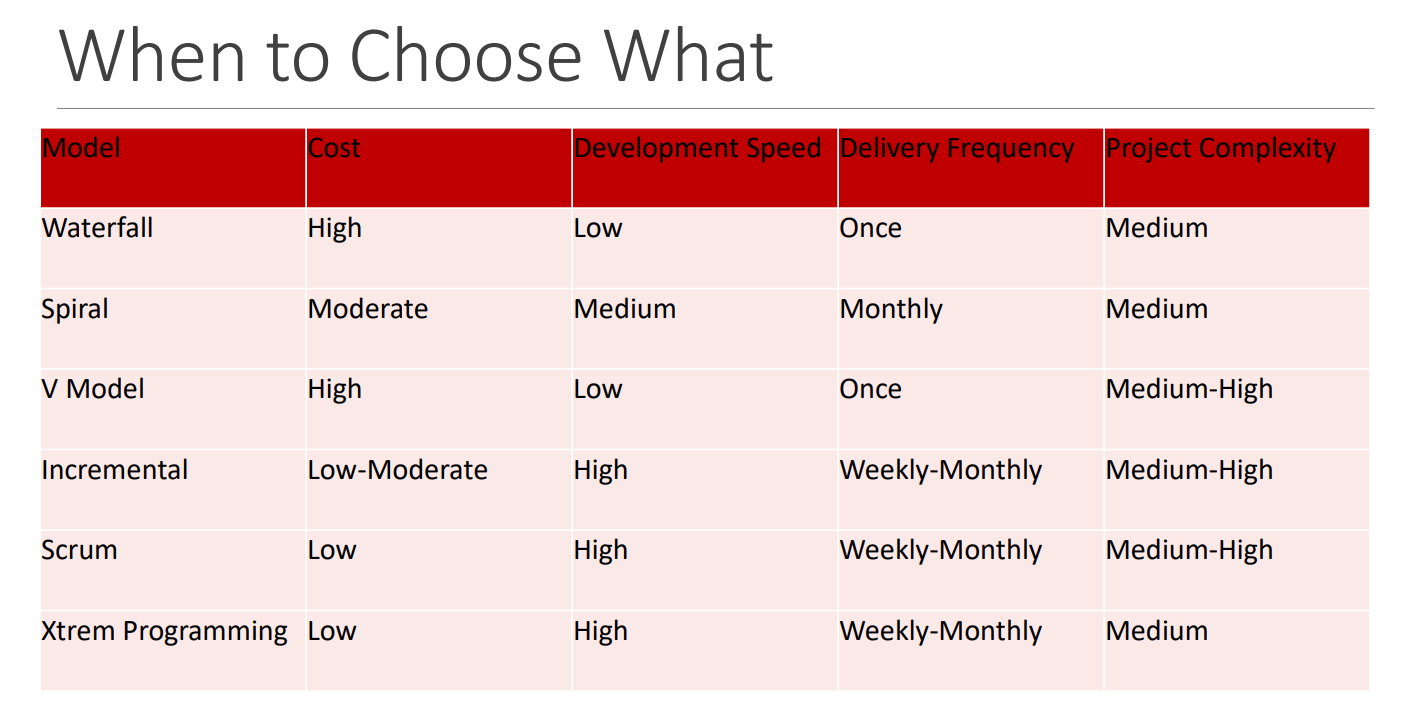
git diff: Workspace and Staging area

git diff --staged: Staging area and local workspace (i.e. last commit)

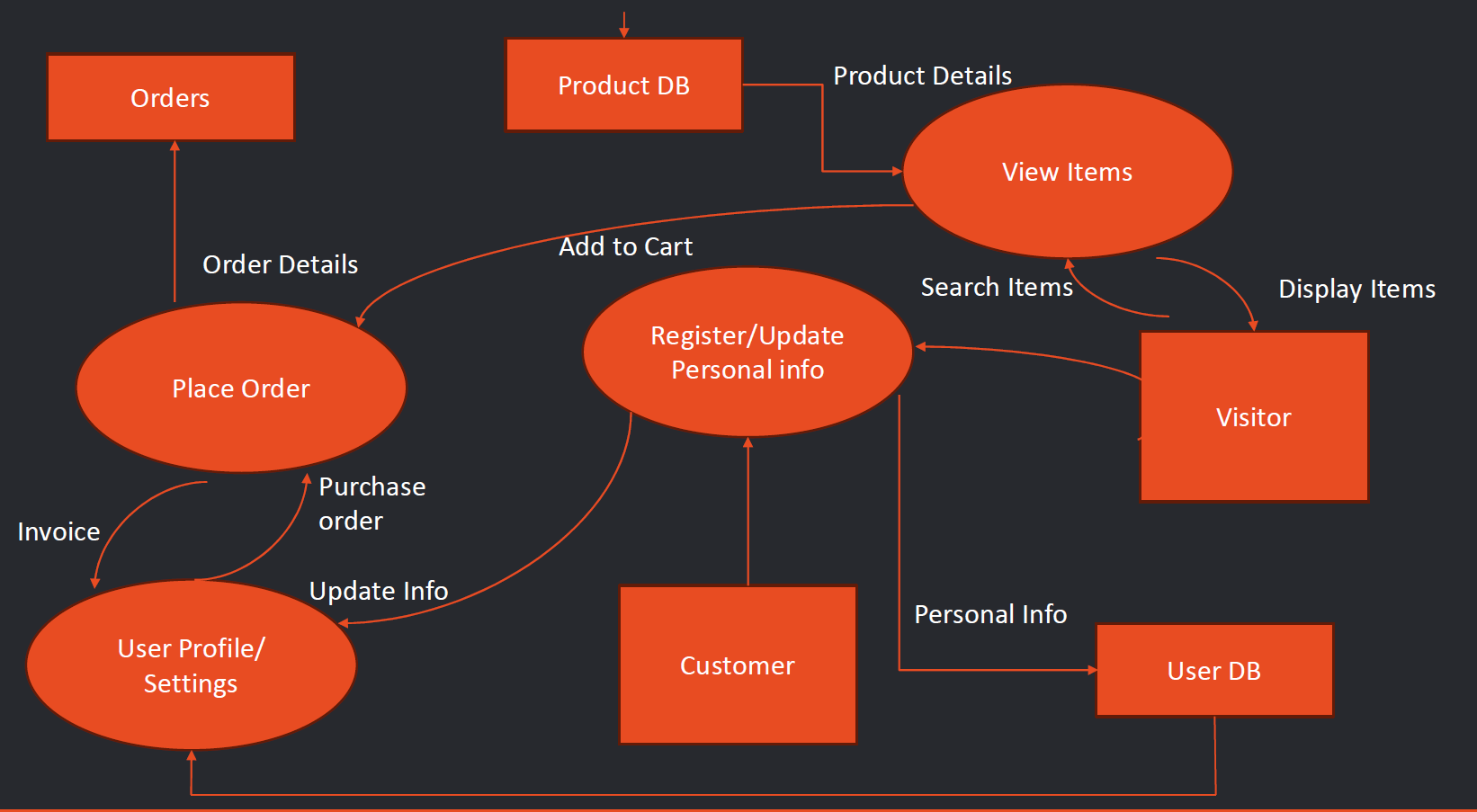
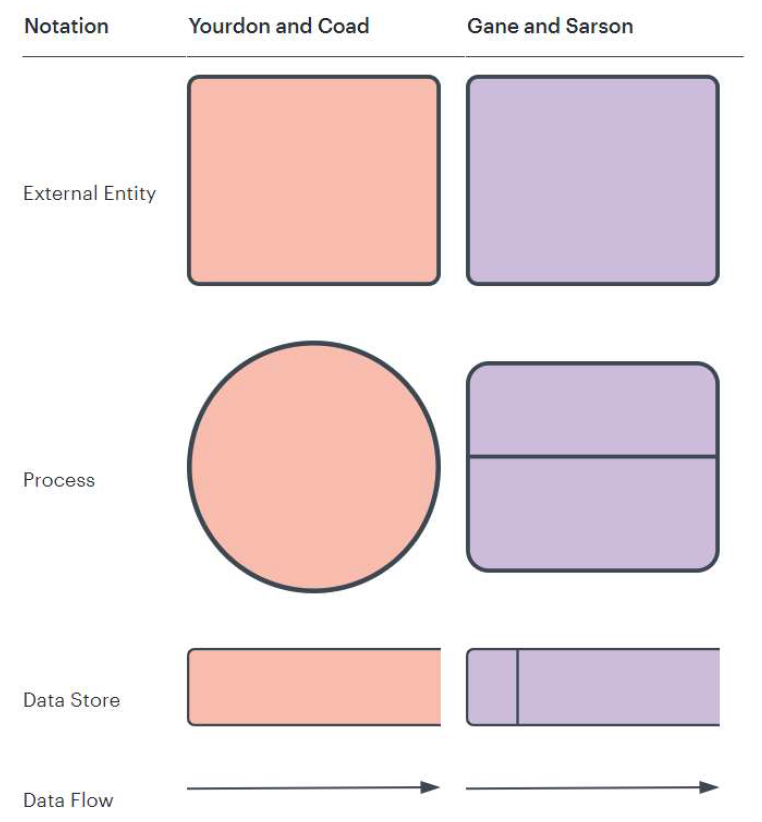
git diff <commit\_hash\_1> <commit\_hash\_2>: Changes between two commits

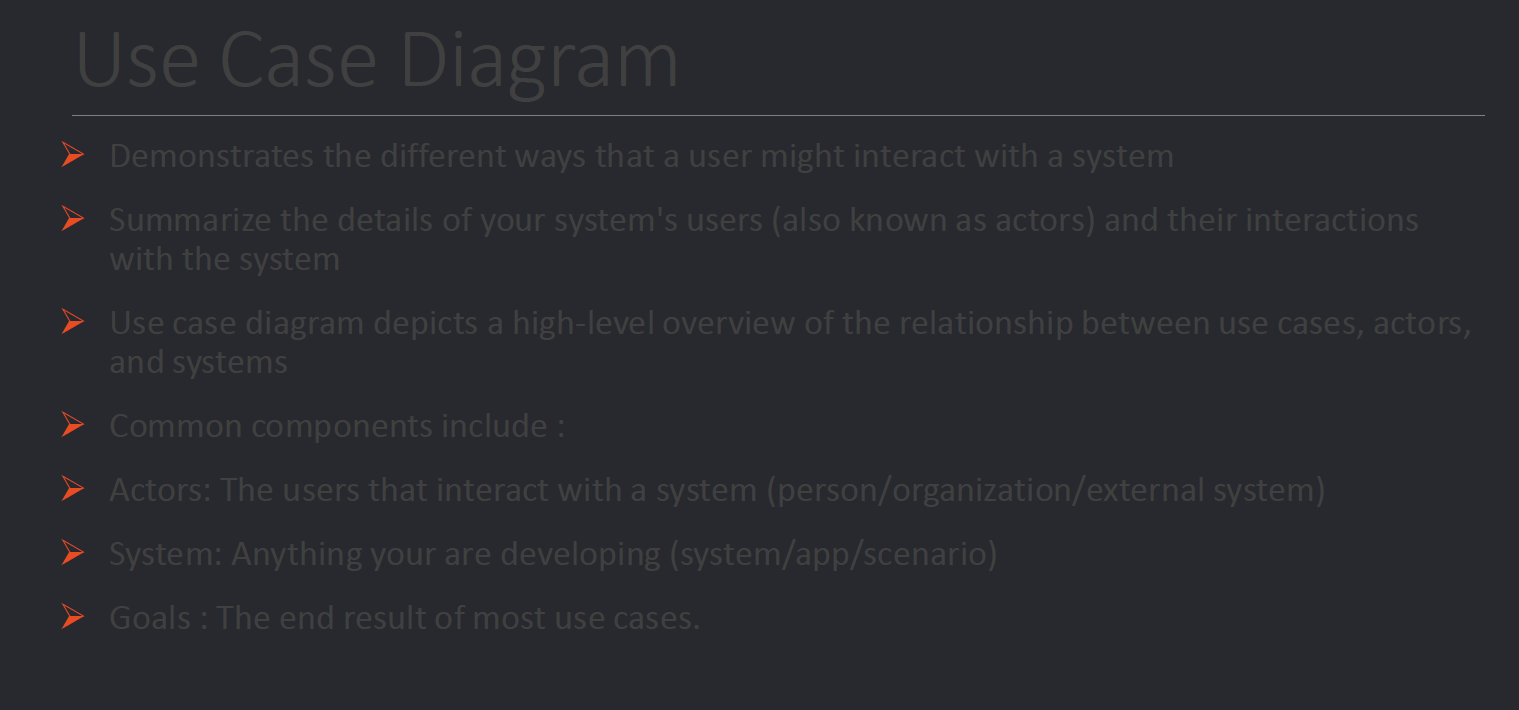
git diff <branch name>: Compare current branch with branch name

SDLC

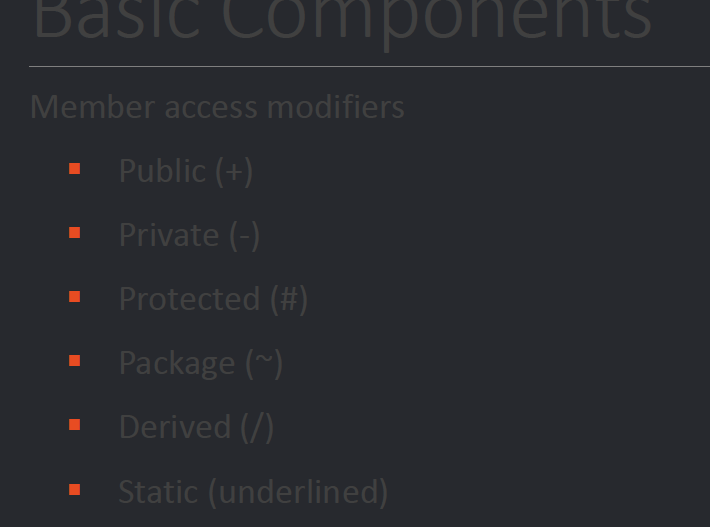


DFD (Data Flow Diagram)

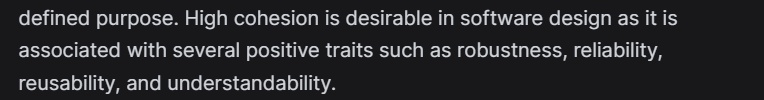




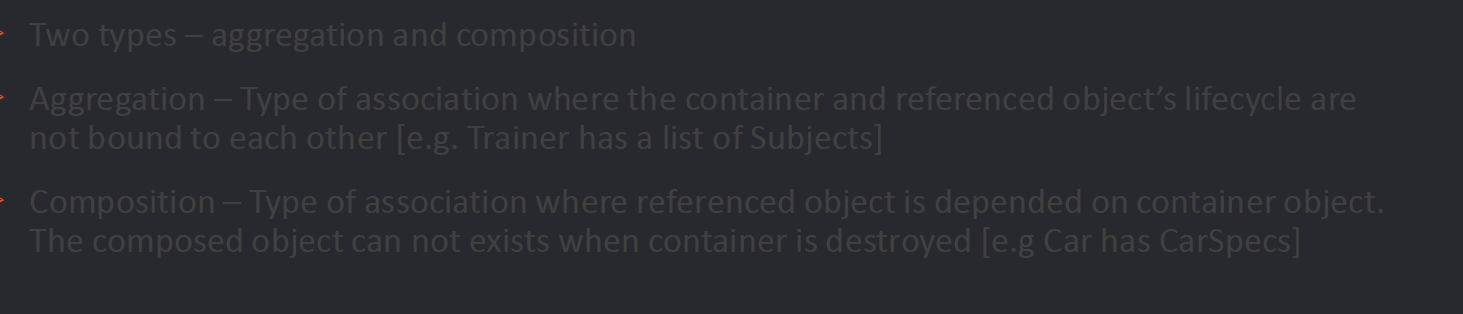
Class Diagrams

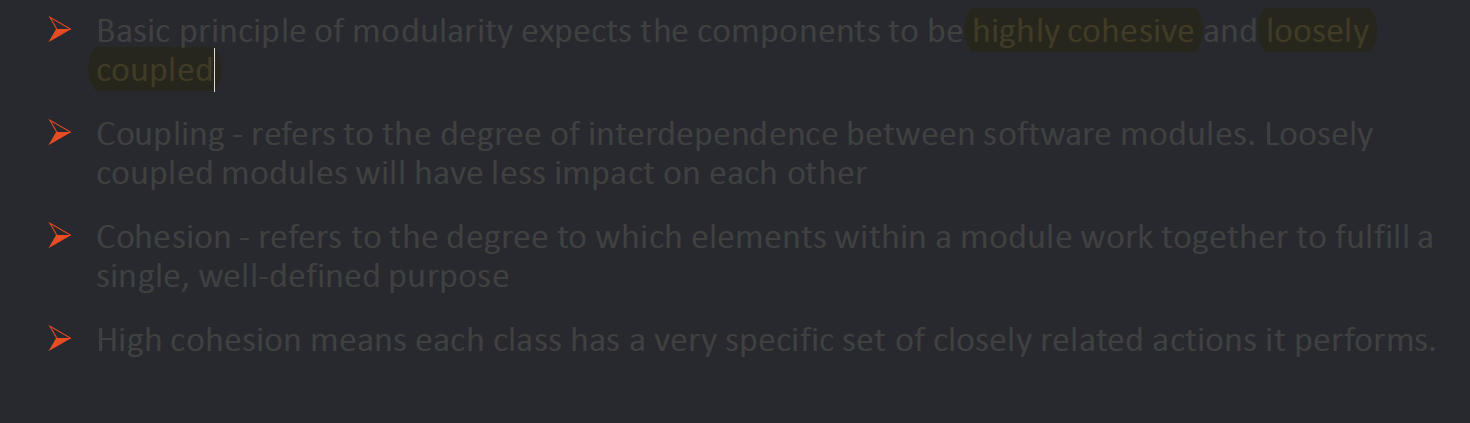


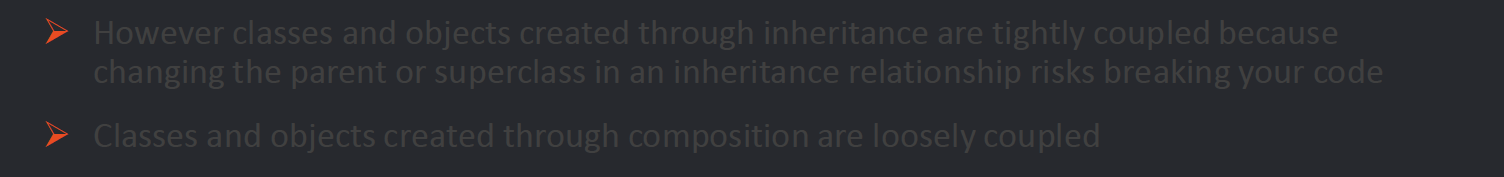


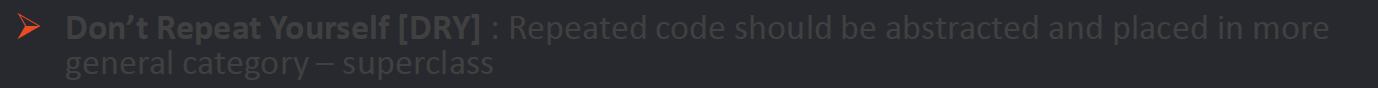


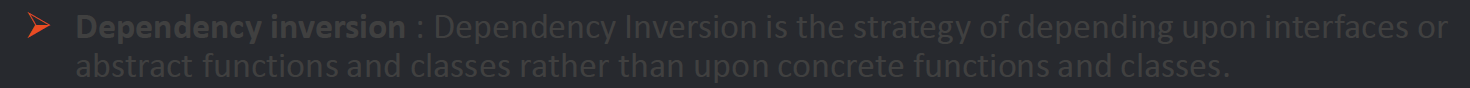
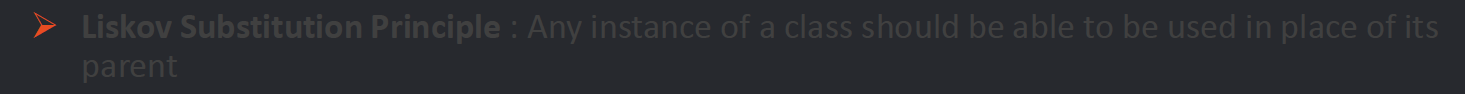
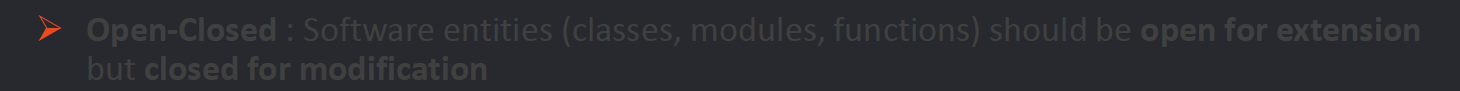
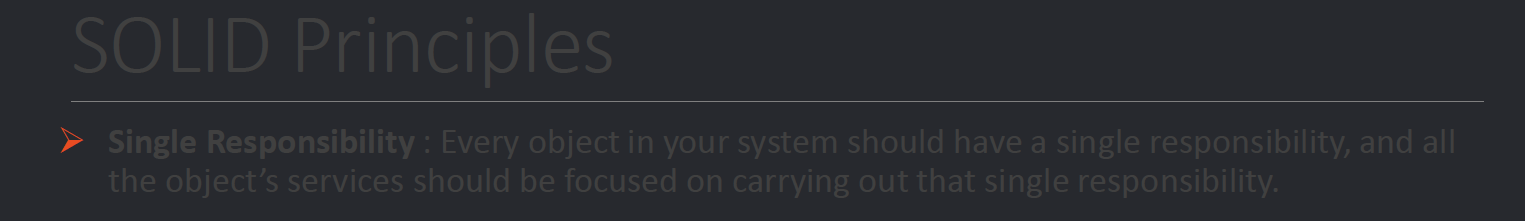
Association

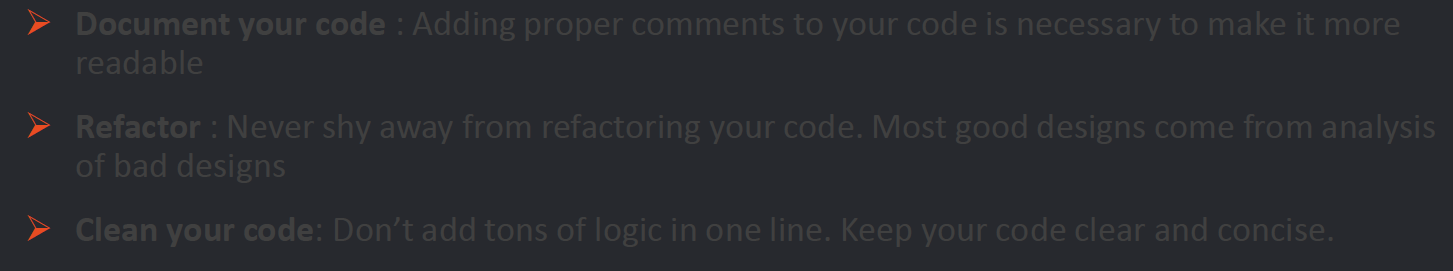
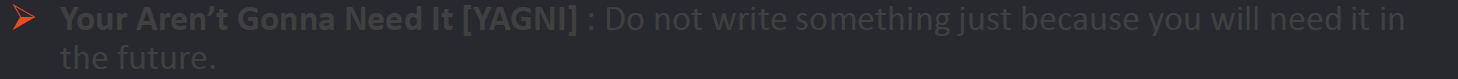




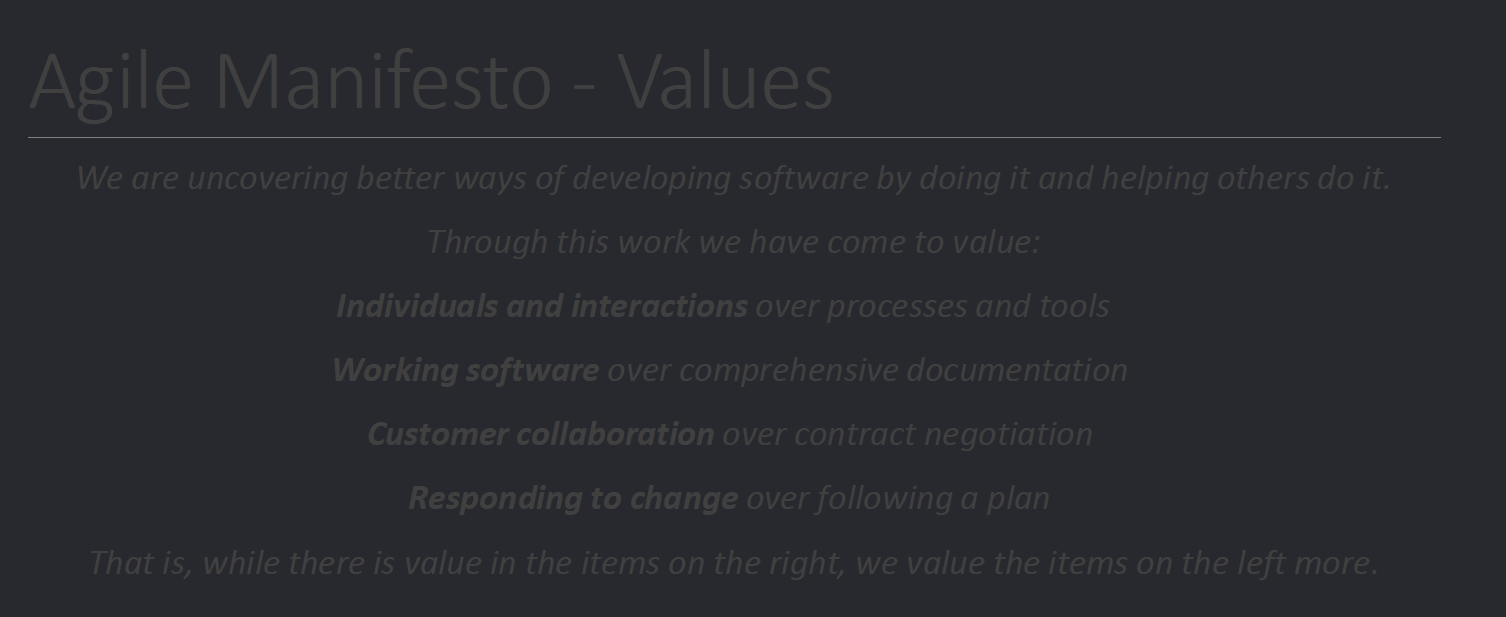


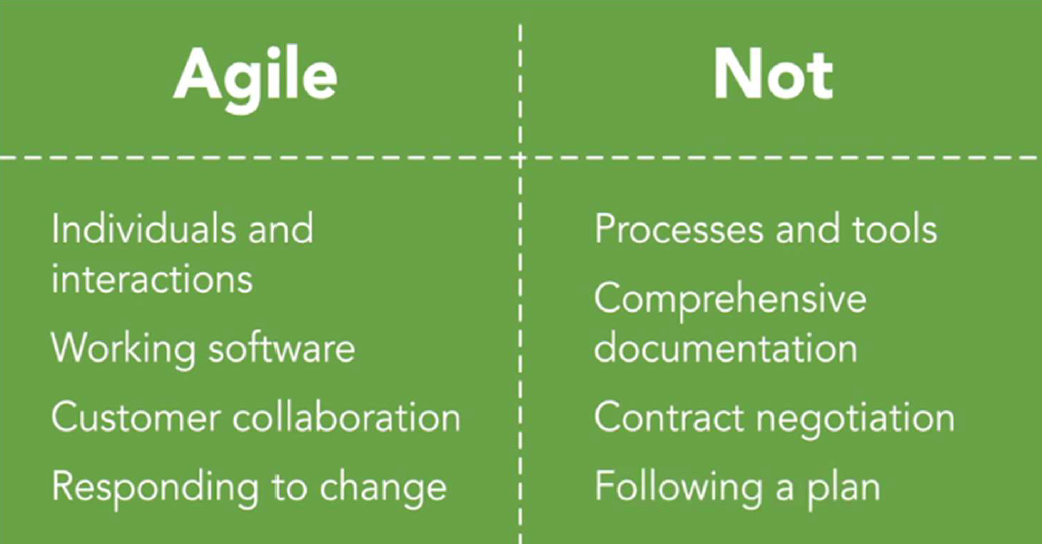


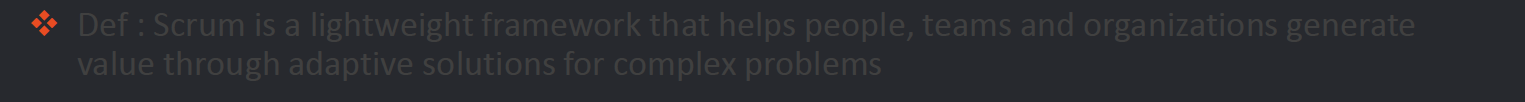


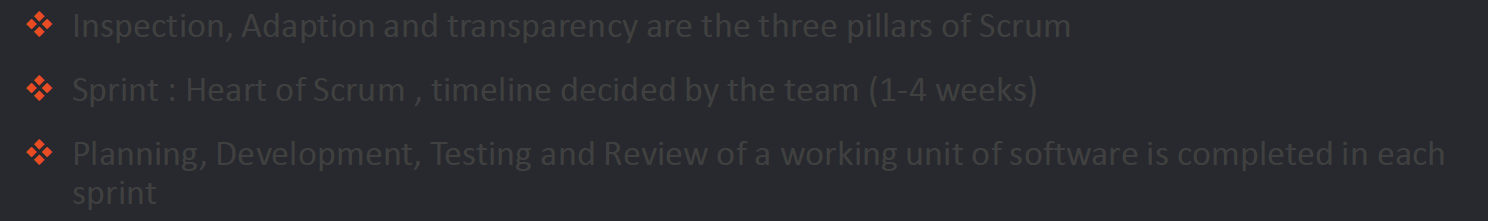


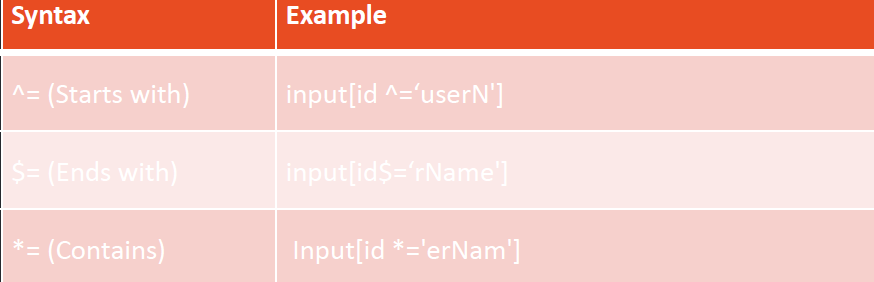
Agile



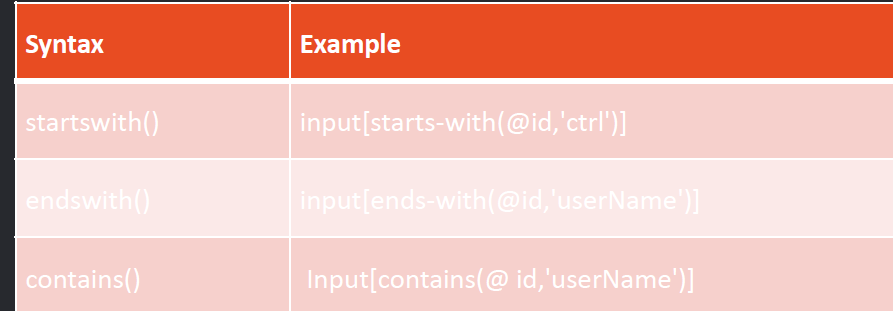


SCRUM  






Using xpath



<https://www.sanfoundry.com/software-engg-mcqs-functional-non-functional-requirements/>

mcq

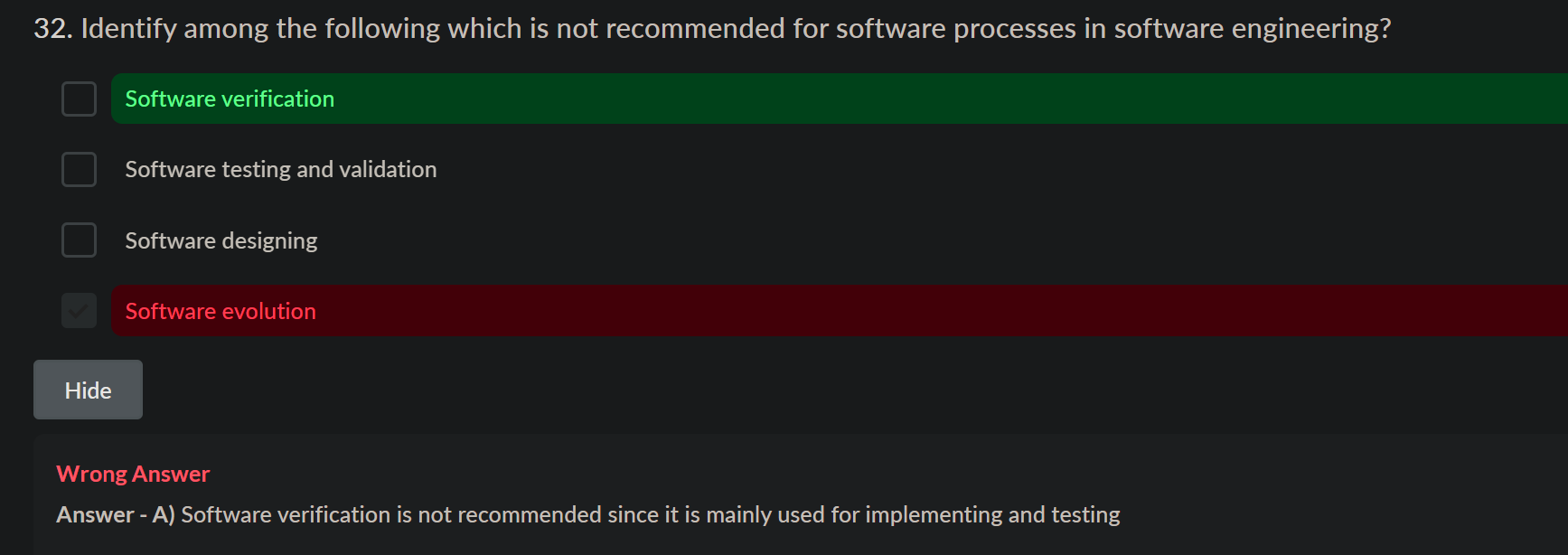


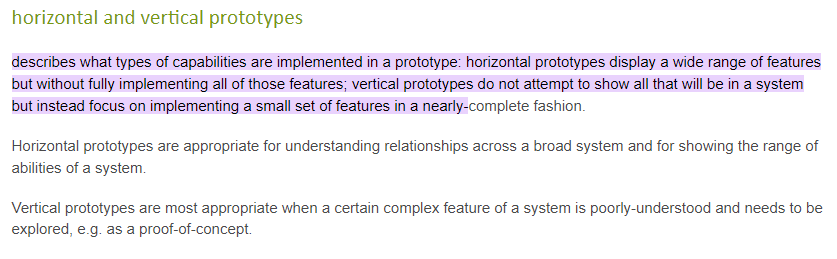


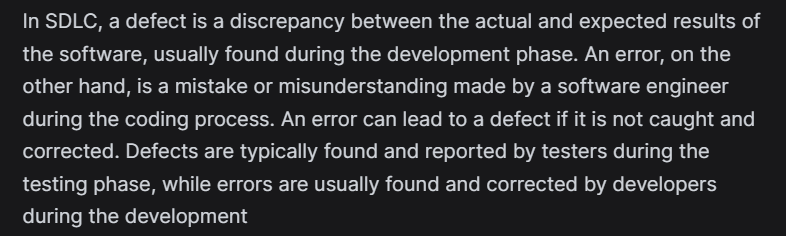


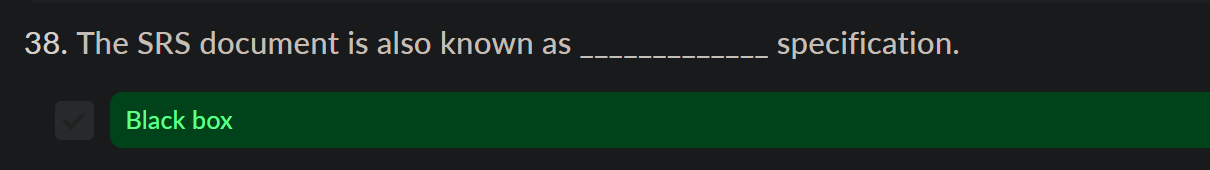


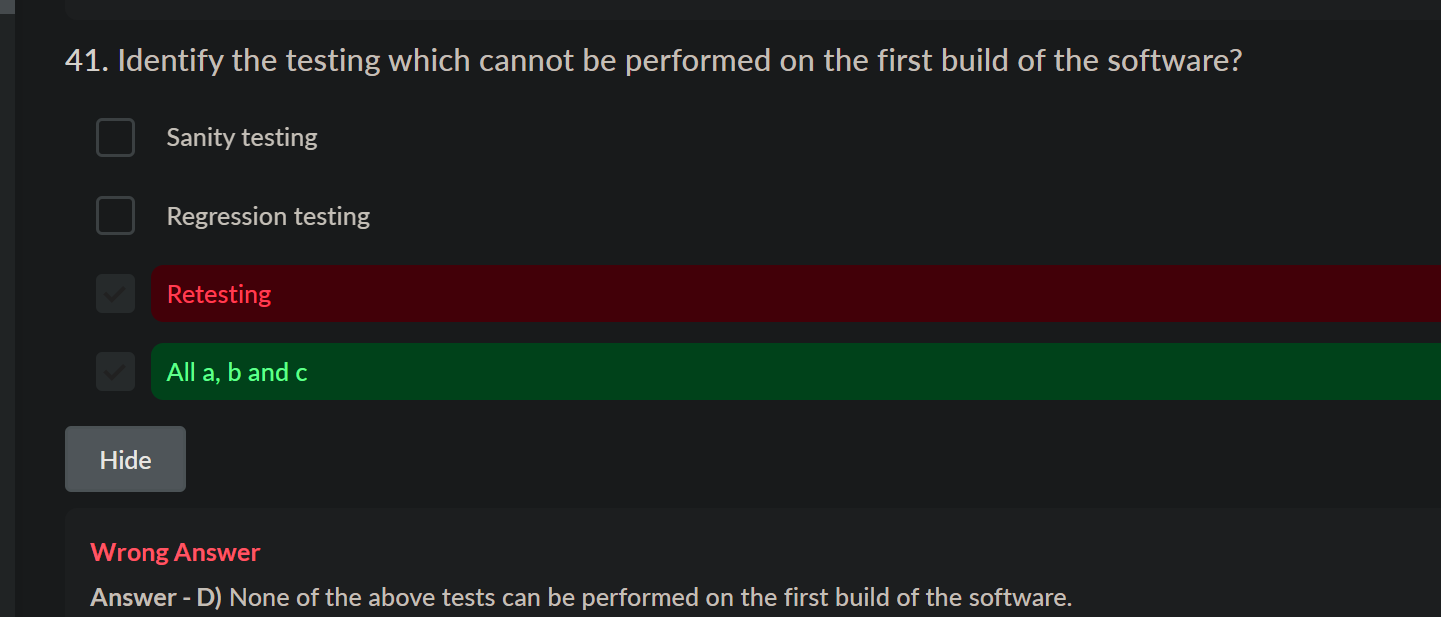
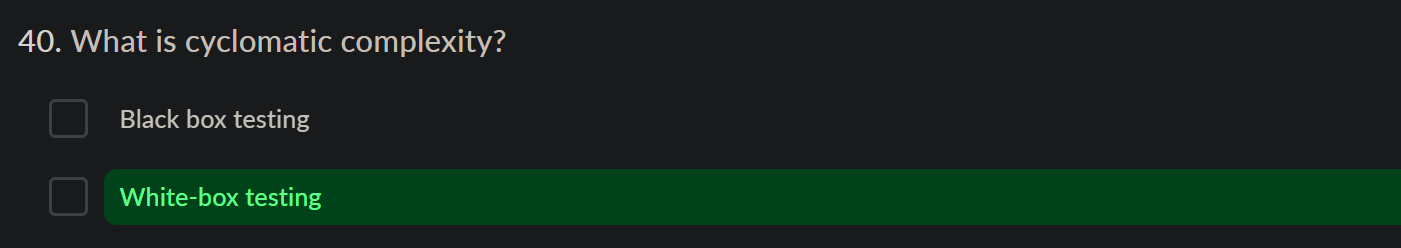
<https://www.interviewbit.com/sdlc-mcq/>

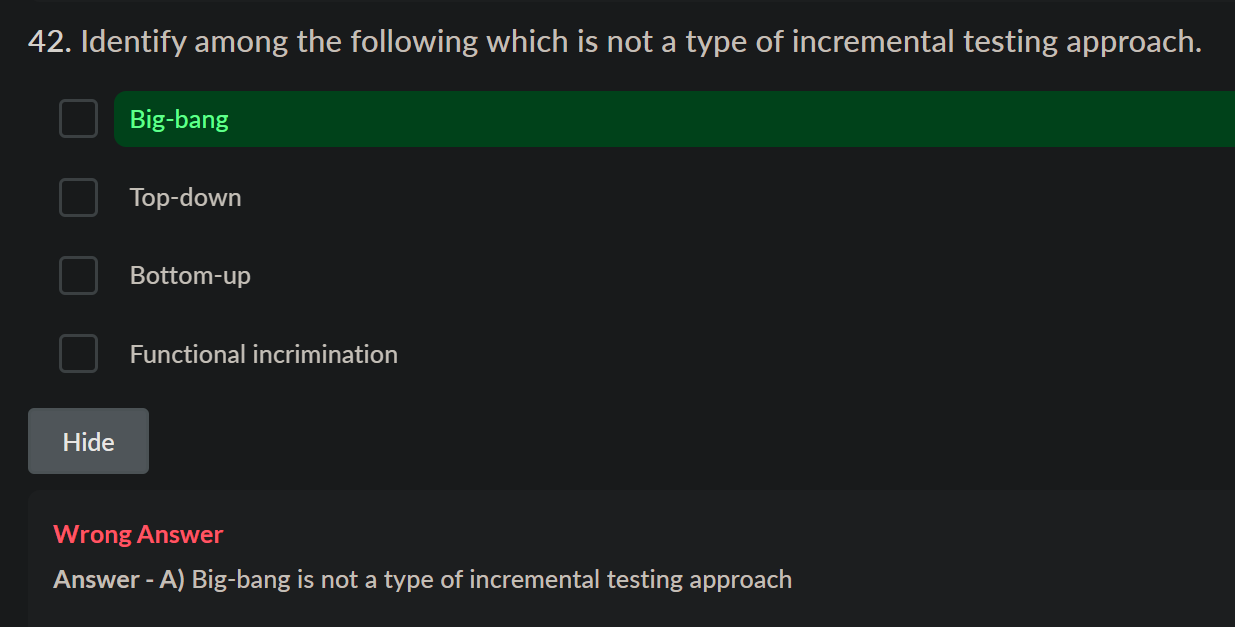


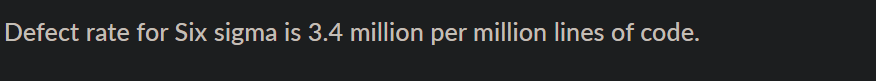
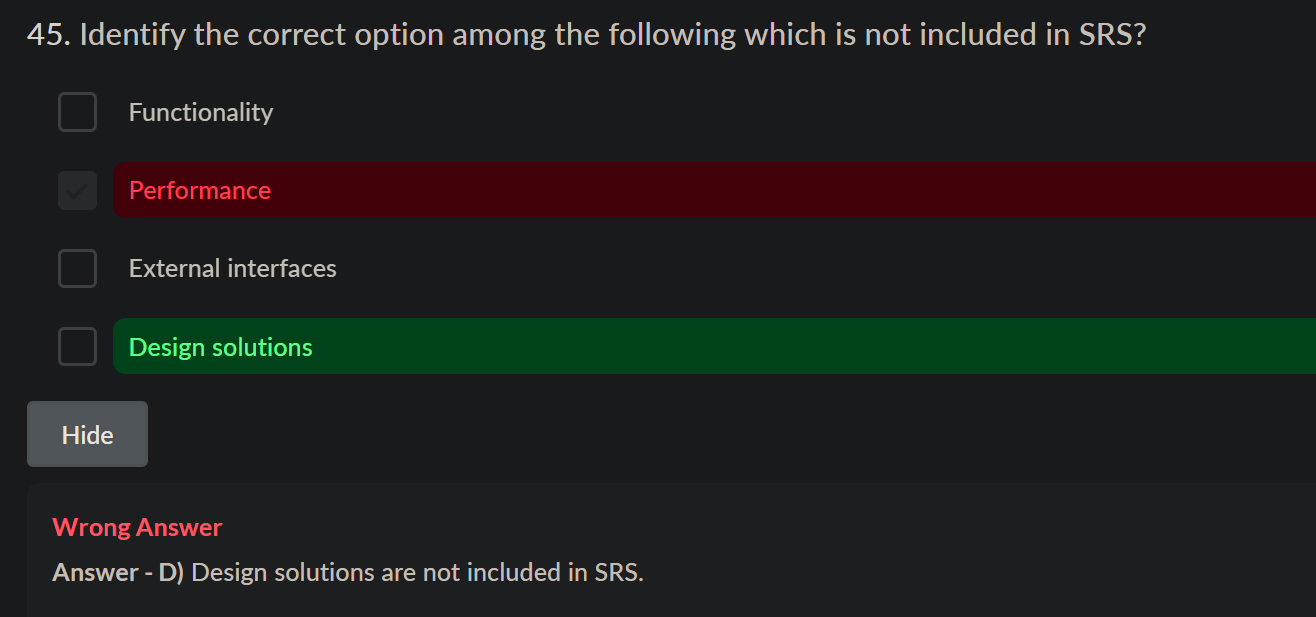




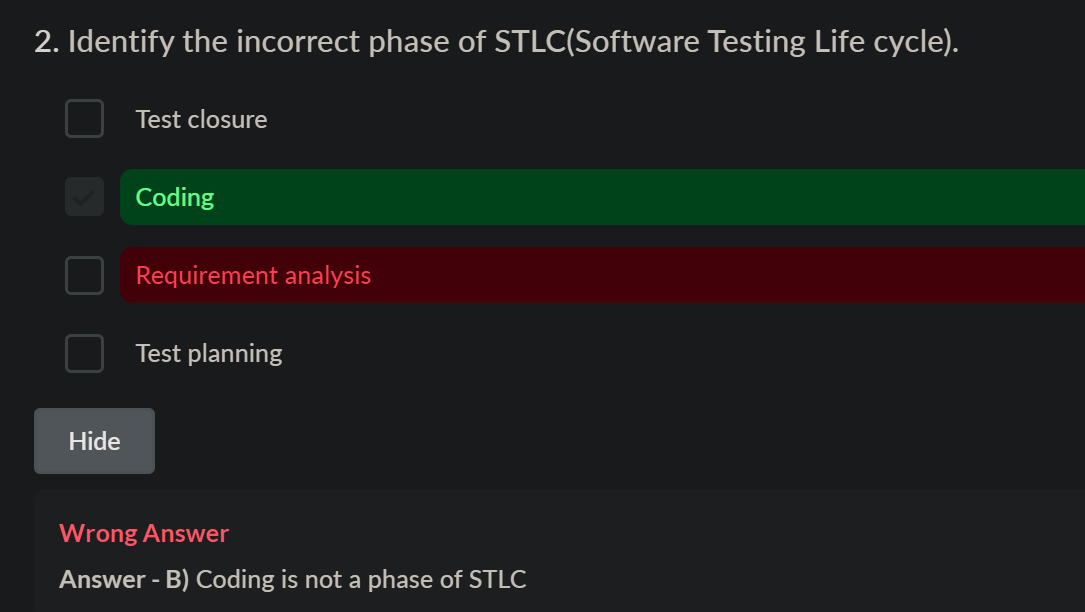






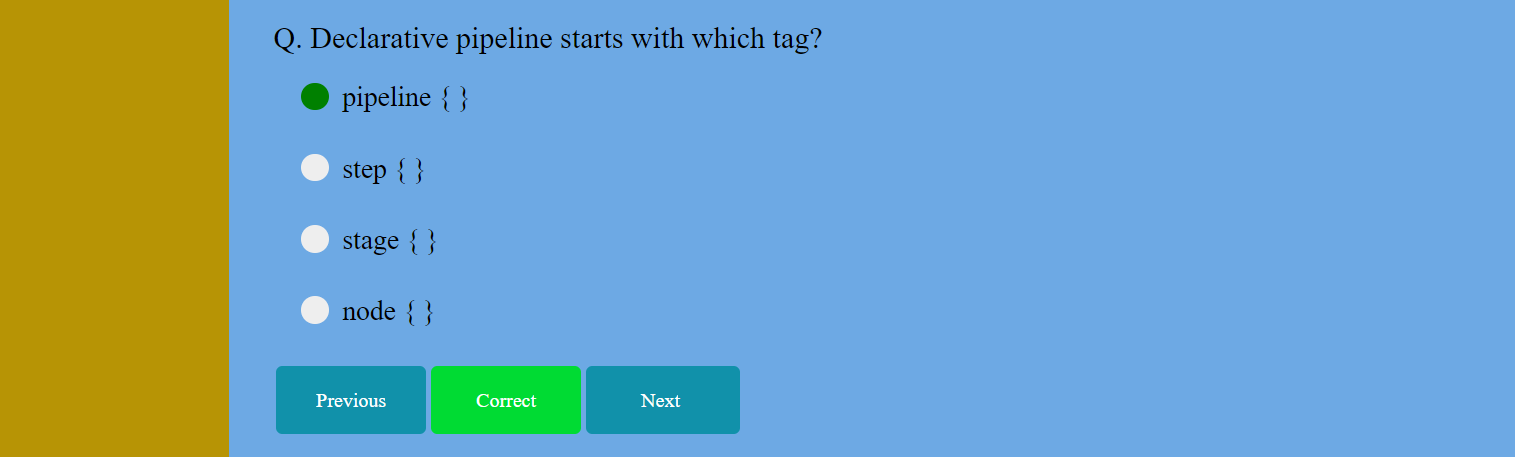


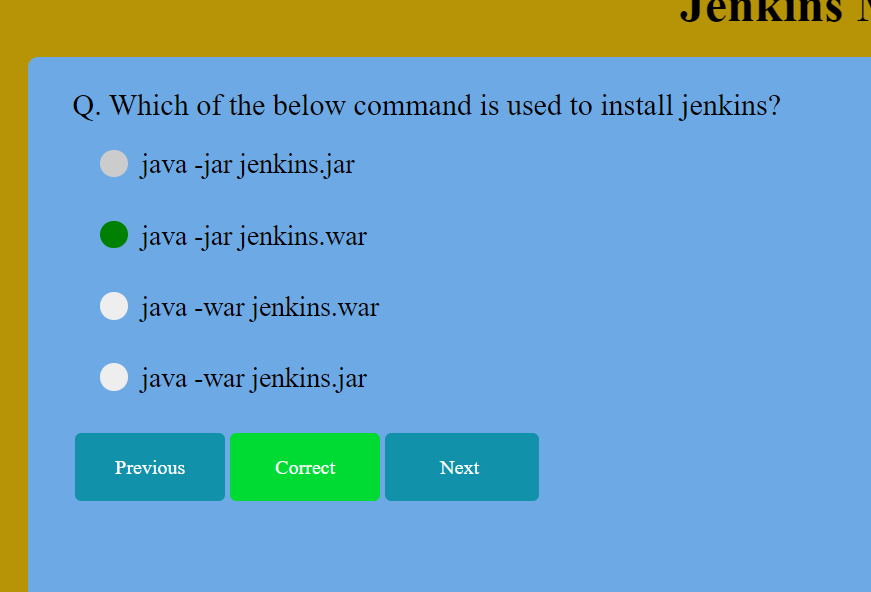




Jenkins

<https://britq.com/jenkins/22>





Networking

