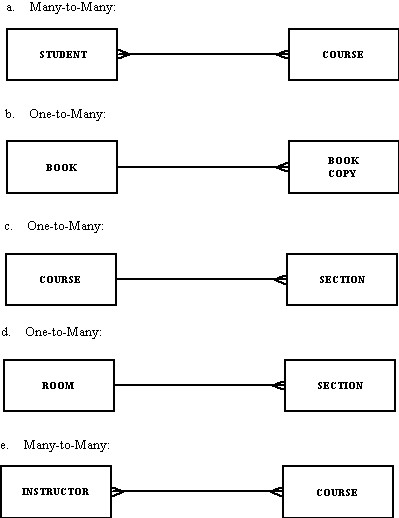
52016/2017 Database

1a.



1b. *Data from driver’s license bureau:*

* 1. Driver’s name, address, and birthdate: is a data--- structured data
  2. The fact that the driver’s name is a 30-character field: metadata; fact describing property
  3. A photo image of the driver: Data; unstructured data
  4. An image of the driver’s fingerprint: Data; unstructured data
  5. The make and serial number of the scanning device that was used to scan the fingerprint: Data; structured data
  6. The resolution (in megapixels) of the camera that was used to photograph the driver: metadata; fact describing context
  7. The fact that the driver’s birth date must precede today’s date by at least 16 years: metadata; fact describing context.

1c. i. The following entities will be needed: Payment, Invoice, Order, Order Line, Product, Customer

ii. All of the above entities will be needed for the SQL query.

Don’t know the year, look at the pq

3a. Match the following terms to the appropriate definitions.

f,e,a,j,g,d,h,I,c,b

Three types of anomalies in tables:

a. Insertion anomaly: A new row cannot be inserted unless all primary key values are

Three types of anomalies in tables:

a. Insertion anomaly: A new row cannot be inserted unless all primary key values are

Three types of anomalies in tables:

a. Insertion anomaly: A new row cannot be inserted unless all primary key values are

Three types of anomalies in tables:

a. Insertion anomaly: A new row cannot be inserted unless all primary key values are

3b. Three types of anomaly:

1. Insertion Anomaly: A new row cannot be inserted unless all primary key values are supplied, leading to the need to insert unnecessary data to accomplish the original goal.
2. Deletion anomaly: Deleting a row results in the loss of important information not stored elsewhere.
3. Modification Anomaly: A simple update must be applied to multiple rows.
4. Demonstrate anomaly types with an example.
5. An insertion anomaly occurs when it is impossible to add an instance of one entity
6. without also adding an instance of another entity (or instances of other entities). For
7. example, if we assume that an ORDER relation includes also information regarding all
8. PRODUCTs that are included in the ORDER, it will be impossible to add a new
9. Demonstrate anomaly types with an example.
10. An insertion anomaly occurs when it is impossible to add an instance of one entity
11. without also adding an instance of another entity (or instances of other entities). For
12. example, if we assume that an ORDER relation includes also information regarding all
13. PRODUCTs that are included in the ORDER, it will be impossible to add a new

3c. the process of designing well-structured relations

Normalization is a formal process for deciding which attributes should be grouped together in a relation so that all anomalies are removed.

Normalization is the process of successively reducing(decomposes) relations with anomalies to produce smaller, well-structured relations.

\*\*\*The process used to minimize data redundancy and dependency in a relational database.

3d. THE PROBLEMS ARE:

Synonyms: Two (or more) attributes that have different names but the same meaning.

Solution: One possible resolution would be to standardize on one of the two attribute names, such as StudentID. Another option is to use a new attribute name, such as StudentNo, to replace both synonyms.

Homonym: An attribute that may have more than one meaning.

Solution: To resolve this conflict, we would probably need to create new attribute names.

Transitive dependency: Merging relations produces transitive dependencies.

Solution: Create 3NF relations by removing the transitive dependency.

Supertype/subtype: May be implied by content of existing relations.

Solution: Create new relations that explicitly recognize this relationship

Supertype/subtype: May be implied by content of existing relations.

Solution: Create new relations that explicitly recognize this relationship.

2017/2018------------------CSC 431

1a. The scalability of a system reflects its ability to deliver high-quality service as demands on the system increase. The three dimensions of scalability are size, distribution, and manageability. OR

Scalability is the ability of the system to cope with increasing numbers of users without reducing the overall quality of service that is delivered to any user.

* Changing the size of a system may involve either scaling up or scaling out.
* Scaling up means replacing resources in the system with more powerful resources. For example, you may increase the memory in a server from 16 Gb to 64 Gb.
* Scaling out means adding more resources to the system (e.g., an extra web server to work alongside an existing server). Scaling out is often more cost-effective than scaling up, especially now that cloud computing makes it easy to add or remove servers from a system.

1b. Distributed systems are inherently more complex than centralized systems. This makes them more difficult to design, implement, and test. It is harder to understand the emergent properties of distributed systems because of the complexity of the interactions between system components and system infrastructure. For example, rather than being dependent on the execution speed of one processor, system performance depends on network bandwidth, network load, and the speed of other computers that are part of the system. Moving resources from one part of the system to another can significantly affect the system’s performance.

1C. In this case, I would chose a fat client model with company information located on a central server (this is critical information and its important that it is consistent for all dealers). Simulations would run on the dealer’s computer as these are used in different ways depending on the individual dealers. A fat client architecture is required because simulations require considerable processing and it would place an unacceptable load on the server if several dealers started simulations at the same time.

2a.

1. Transparency To what extent should the distributed system appear to the user as a single system? When is it useful for users to understand that the system is distributed?

2. Openness Should a system be designed using standard protocols that support interoperability, or should more specialized protocols be used? Although standard network protocols are now universally used, this is not the case for higher levels of interaction, such as service communication.

3. Scalability How can the system be constructed so that it is scalable? That is, how can the overall system be designed so that its capacity can be increased in response to increasing demands made on the system?

4. Security How can usable security policies be defined and implemented that apply across a set of independently managed systems?

5. Quality of service How should the quality of service that is delivered to system users be specified, and how should the system be implemented to deliver an acceptable quality of service to all users.

6. Failure management How can system failures be detected, contained (so that they have minimal effects on other components in the system), and repaired?

2b. In a remote procedure call, an executing component on one computer (A) calls a procedure or method, which is part of a component that is executing on a different computer (B). The role of the middleware is to coordinate this interaction. There are several steps involved in this: 1. The provision of a stub procedure with the same interface as the called component. Calling this stub procedure initiates a call to the system middleware. 2. The middleware running on computer A accepts the call and discovers the location of the called component. 3. It translates the parameters into a standard format and sends these to computer B along with a request to call the required component. 4. The middleware on computer B converts the parameters into the appropriate format for the language of the called component and then calls that component. 5. After execution, the called component returns the result to the middleware on computer B which then translates this into the middleware standard format. 6. The result is transmitted to the middleware on computer A, which then translates that into the appropriate language format and returns it to the original calling component.

OR

Procedural interaction involves one compute calling on a known service offered by some other computer and waiting for that service to be delivered. In remote procedural call (RPC) one component calls another component as if it was a local procedure. The middleware in the system intercepts this call and passes it to a remote component. This carries out the required computation and, via the middleware, returns the result to the calling procedure.  
  
Let us consider an example of RPC:  
  
A man goes to restaurant. He picks up the menu and selects his lunch choice. The waiter writes-down his order along with others order and retreats to the kitchen, leaving the order with the cook. The cook prepares the meal and the waiter delivers it to the man. This can be comparable to components interacting in a software system where one component calls methods from other components

2c.

Deploying software as a service has the potential for reducing the IT support costs as there is no need to install and support separate software on each client. Rather, all software is hosted on a server and when e.g. upgrades are required, only the server (or servers) need be upgraded. There are no support problems with different computers in an organization running different software versions. General help support is provided by the service provider rather than the local IT staff. The additional costs that can arise from this model are: 1. Network costs, as obviously there is a considerable increase in network traffic. Service providers (such as Amazon) may charge for data uploads and downloads. This is only applicable of the service is provided by a 3rd party rather than in-house. 2. Server costs, as the servers are responsible for all computation and so must either be more powerful or more numerous. This is most significant if the service is provided in-house. 3. There may in fact be additional support costs from this model in the short term if it requires users to change the software that they normally use. This is likely to lead to additional demands for help.

3a.

1. Architecture in the small is concerned with the architecture of individual programs. At this level, we are concerned with the way that an individual program is decomposed into components. This chapter is mostly concerned with program architectures.

2. Architecture in the large is concerned with the architecture of complex enterprise systems that include other systems, programs, and program components. These enterprise systems may be distributed over different computers, which may be owned and managed by different companies.

3b.  [YouTube](https://en.wikipedia.org/wiki/YouTube), [Netflix](https://en.wikipedia.org/wiki/Netflix), [Hulu](https://en.wikipedia.org/wiki/Hulu), [Vudu](https://en.wikipedia.org/wiki/Vudu" \o "Vudu), [Amazon Prime Video](https://en.wikipedia.org/wiki/Prime_Video), [DirecTV](https://en.wikipedia.org/wiki/DirecTV), [SlingTV](https://en.wikipedia.org/wiki/Sling_TV" \o "Sling TV)

Client-server model is the appropriate architectural pattern for such kind of system. Netflix stores all the movies they sell in a database where the client can search these movies by genre, movie title etc., all via web-based interface. Also, movies can be downloaded and paid accordingly. Then, the server manages the movies ordering also via web-based interface.

3bi.

Buy

Login authentication system

Register System

Browse

Sell/distribute

Download

Logout system

3c.

Organizes the system into layers, with related functionality associated with each layer. A layer provides services to the layer above it, so the lowest level layers represent core services that are likely to be used throughout the system.

an example of a layered architecture with four layers. The lowest layer includes system support software—typically, database and operating system support. The next layer is the application layer, which includes the components concerned with the application functionality and utility components used by other application components. The third layer is concerned with user interface management and providing user authentication and authorization, with the top layer providing user interface facilities. Of course, the number of layers is arbitrary.

User interface

User interface management Authentication and authorization

Core business logic/application functionality System utilities

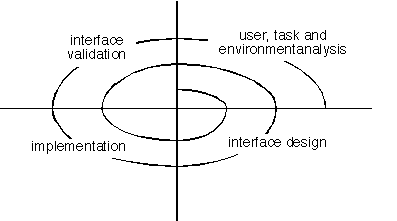
System support (OS, database, etc.)

System support (OS, database, etc.)

System support (OS, database, etc.)

4a. The user interface analysis and design process encompasses four distinct framework activities:

1. User, task and environment analysis and modeling.
2. Interface design
3. Interface construction (implementation)
4. Interface validation



a. User, task, environmental analysis, and modeling: Initially, the focus is based on the profile of users who will interact with the system, i.e. understanding, skill and knowledge, type of user, etc, based on the user’s profile users are made into categories. From each category requirements are gathered. Based on the requirements developer understand how to develop the interface. Once all the requirements are gathered a detailed analysis is conducted. In the analysis part, the tasks that the user performs to establish the goals of the system are identified, described and elaborated. The analysis of the user environment focuses on the physical work environment. Among the questions to be asked are:

* Where will the interface be located physically?
* Will the user be sitting, standing, or performing other tasks unrelated to the interface?
* Does the interface hardware accommodate space, light, or noise constraints?
* Are there special human factors considerations driven by environmental factors?

b. Interface Design: The goal of this phase is to define the set of interface objects and actions i.e. Control mechanisms that enable the user to perform desired tasks. Indicate how these control mechanisms affect the system. Specify the action sequence of tasks and subtasks, also called a user scenario. Indicate the state of the system when the user performs a particular task. Always follow the three golden rules stated by Theo Mandel. Design issues such as response time, command and action structure, error handling, and help facilities are considered as the design model is refined. This phase serves as the foundation for the implementation phase.

c. Interface construction and implementation: The implementation activity begins with the creation of prototype (model) that enables usage scenarios to be evaluated. As iterative design process continues a User Interface toolkit that allows the creation of windows, menus, device interaction, error messages, commands, and many other elements of an interactive environment can be used for completing the construction of an interface.

d. Interface Validation: This phase focuses on testing the interface. The interface should be in such a way that it should be able to perform tasks correctly and it should be able to handle a variety of tasks. It should achieve all the user’s requirements. It should be easy to use and easy to learn. Users should accept the interface as a useful one in their work.

4b. Golden Rules:

The following are the golden rules stated by Theo Mandel that must be followed during the design of the interface.

Place the user in control:

* Define the interaction modes in such a way that does not force the user into unnecessary or undesired actions: The user should be able to easily enter and exit the mode with little or no effort.
* Provide for flexible interaction: Different people will use different interaction mechanisms, some might use keyboard commands, some might use mouse, some might use touch screen, etc, Hence all interaction mechanisms should be provided.
* Allow user interaction to be interruptable and undoable: When a user is doing a sequence of actions the user must be able to interrupt the sequence to do some other work without losing the work that had been done. The user should also be able to do undo operation.
* Streamline interaction as skill level advances and allow the interaction to be customized: Advanced or highly skilled user should be provided a chance to customize the interface as user wants which allows different interaction mechanisms so that user doesn’t feel bored while using the same interaction mechanism.
* Hide technical internals from casual users: The user should not be aware of the internal technical details of the system. He should interact with the interface just to do his work.
* Design for direct interaction with objects that appear on screen: The user should be able to use the objects and manipulate the objects that are present on the screen to perform a necessary task. By this, the user feels easy to control over the screen.

Reduce the user’s memory load:

* Reduce demand on short-term memory: When users are involved in some complex tasks the demand on short-term memory is significant. So the interface should be designed in such a way to reduce the remembering of previously done actions, given inputs and results.
* Establish meaningful defaults: Always initial set of defaults should be provided to the average user, if a user needs to add some new features then he should be able to add the required features.
* Define shortcuts that are intuitive: Mnemonics should be used by the user. Mnemonics means the keyboard shortcuts to do some action on the screen.
* The visual layout of the interface should be based on a real-world metaphor: Anything you represent on a screen if it is a metaphor for real-world entity then users would easily understand.
* Disclose information in a progressive fashion: The interface should be organized hierarchically i.e. on the main screen the information about the task, an object or some behavior should be presented first at a high level of abstraction. More detail should be presented after the user indicates interest with a mouse pick.

Make the interface consistent:

* Allow the user to put the current task into a meaningful context: Many interfaces have dozens of screens. So it is important to provide indicators consistently so that the user know about the doing work. The user should also know from which page has navigated to the current page and from the current page where can navigate.
* Maintain consistency across a family of applications: The development of some set of applications all should follow and implement the same design, rules so that consistency is maintained among applications.
* If past interactive models have created user expectations do not make changes unless there is a compelling reason.

4c. Abeg do am yourself.2

5a.

i. Functional concerns which reflect specific functionality required. The base system should implement core functionality and extensions, implemented as aspects, can implement secondary functionality.

ii. Quality of service concerns related to non-functional behaviour of the system. Aspects may be used to implement cross-cutting functionality, such as a cache, which helps these requirements to be met.

iii. Policy concerns relating to the overall policies of use of the system. These are inevitably cross-cutting. Aspects may be used to implement these concerns.

iv. System concerns which relate to the attributes of the system as a whole. Aspects may be used to implement monitoring that checks the system attributes.

v. Organisational concerns that are related to organisational goals and priorities such as maintaining reputation. Aspects have limited usefulness in implementing this type of concern.

5b. Tangling arises when one module implements (part of) several requirements; scattering arises where the implementation of a single requirement is spread across several modules.

2nd part

Say several requirements in a system have a shared component that implements the issuing of a confirmation to a user that some action has occurred. This is tangled system. If one of these changes so that some change to the confirmation is required, all of the other requirements that issue confirmations are affected by this.

If the implementation of each requirement which issues a confirmation is scattered across several modules, then all of these have to be checked that the change will not affect their operation.

c. join point: An event in an executing program where the advice

associated with an aspect may be executed.

Pointcut: A statement, included in an aspect, that defines the join

points where the associated aspect advice should be

executed. In this example, the pointcut is a simple statement:

*before: call (public void update\* (..))*

2016/17 -----419

1a.

In a fat-client system, some of the application processing is carried out on the client

whereas in a thin client system only the user interface is displayed on the client and

all of the application processing is carried out on the server. However, modern web

browsers are all javascript enabled which means that code can be downloaded from

the web page on the server and executed within the client browser. This means that

some of the functionality of fat clients can be replicated without the need to install

software on the client system.

2a.

1. Transparency To what extent should the distributed system appear to the user as a single system? When is it useful for users to understand that the system is distributed?

2. Openness Should a system be designed using standard protocols that support interoperability, or should more specialized protocols be used? Although standard network protocols are now universally used, this is not the case for higher levels of interaction, such as service communication.

3. Scalability How can the system be constructed so that it is scalable? That is, how can the overall system be designed so that its capacity can be increased in response to increasing demands made on the system?

4. Security How can usable security policies be defined and implemented that apply across a set of independently managed systems?

5. Quality of service How should the quality of service that is delivered to system users be specified, and how should the system be implemented to deliver an acceptable quality of service to all users.

6. Failure management How can system failures be detected, contained (so that they have minimal effects on other components in the system), and repaired?

2b. in the architecture so that in the event of one component failing, you can switch

immediately to a backup component. You also need to have several copies of the

data that is being processed. Security requires minimizing the number of copies of

the data and, wherever possible, adopting an architecture where each component

only knows as much as it needs to, to do its job. This reduces the chance of

intruders accessing the data.

Therefore, there is a fundamental architectural conflict between availability

(replication, several copies) and security (specialization, minimal copies). The

system architect has to find the best compromise between these fundamentally

opposing requirements.

3a. 1. A logical view, which shows the key abstractions in the system as objects or

object classes. It should be possible to relate the system requirements to entities

in this logical view.

2. A process view, which shows how, at run-time, the system is composed of interacting

processes. This view is useful for making judgments about nonfunctional

system characteristics such as performance and availability.

3. A development view, which shows how the software is decomposed for development,

that is, it shows the breakdown of the software into components that are

implemented by a single developer or development team. This view is useful for

software managers and programmers.

4. A physical view, which shows the system hardware and how software components

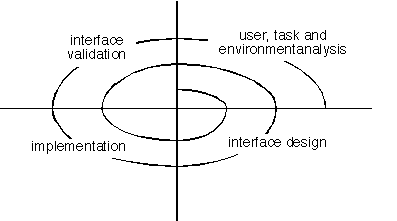
are distributed across the processors in the system. This view is useful for

systems engineers planning a system deployment.

3b.

4a. The user interface analysis and design process encompasses four distinct framework activities:

1. User, task and environment analysis and modeling.
2. Interface design
3. Interface construction (implementation)
4. Interface validation



a. User, task, environmental analysis, and modeling: Initially, the focus is based on the profile of users who will interact with the system, i.e. understanding, skill and knowledge, type of user, etc, based on the user’s profile users are made into categories. From each category requirements are gathered. Based on the requirements developer understand how to develop the interface. Once all the requirements are gathered a detailed analysis is conducted. In the analysis part, the tasks that the user performs to establish the goals of the system are identified, described and elaborated. The analysis of the user environment focuses on the physical work environment. Among the questions to be asked are:

* Where will the interface be located physically?
* Will the user be sitting, standing, or performing other tasks unrelated to the interface?
* Does the interface hardware accommodate space, light, or noise constraints?
* Are there special human factors considerations driven by environmental factors?

b. Interface Design: The goal of this phase is to define the set of interface objects and actions i.e. Control mechanisms that enable the user to perform desired tasks. Indicate how these control mechanisms affect the system. Specify the action sequence of tasks and subtasks, also called a user scenario. Indicate the state of the system when the user performs a particular task. Always follow the three golden rules stated by Theo Mandel. Design issues such as response time, command and action structure, error handling, and help facilities are considered as the design model is refined. This phase serves as the foundation for the implementation phase.

c. Interface construction and implementation: The implementation activity begins with the creation of prototype (model) that enables usage scenarios to be evaluated. As iterative design process continues a User Interface toolkit that allows the creation of windows, menus, device interaction, error messages, commands, and many other elements of an interactive environment can be used for completing the construction of an interface.

d. Interface Validation: This phase focuses on testing the interface. The interface should be in such a way that it should be able to perform tasks correctly and it should be able to handle a variety of tasks. It should achieve all the user’s requirements. It should be easy to use and easy to learn. Users should accept the interface as a useful one in their work.

4b. Golden Rules:

The following are the golden rules stated by Theo Mandel that must be followed during the design of the interface.

Place the user in control:

* Define the interaction modes in such a way that does not force the user into unnecessary or undesired actions: The user should be able to easily enter and exit the mode with little or no effort.
* Provide for flexible interaction: Different people will use different interaction mechanisms, some might use keyboard commands, some might use mouse, some might use touch screen, etc, Hence all interaction mechanisms should be provided.
* Allow user interaction to be interruptable and undoable: When a user is doing a sequence of actions the user must be able to interrupt the sequence to do some other work without losing the work that had been done. The user should also be able to do undo operation.
* Streamline interaction as skill level advances and allow the interaction to be customized: Advanced or highly skilled user should be provided a chance to customize the interface as user wants which allows different interaction mechanisms so that user doesn’t feel bored while using the same interaction mechanism.
* Hide technical internals from casual users: The user should not be aware of the internal technical details of the system. He should interact with the interface just to do his work.
* Design for direct interaction with objects that appear on screen: The user should be able to use the objects and manipulate the objects that are present on the screen to perform a necessary task. By this, the user feels easy to control over the screen.

Reduce the user’s memory load:

* Reduce demand on short-term memory: When users are involved in some complex tasks the demand on short-term memory is significant. So the interface should be designed in such a way to reduce the remembering of previously done actions, given inputs and results.
* Establish meaningful defaults: Always initial set of defaults should be provided to the average user, if a user needs to add some new features then he should be able to add the required features.
* Define shortcuts that are intuitive: Mnemonics should be used by the user. Mnemonics means the keyboard shortcuts to do some action on the screen.
* The visual layout of the interface should be based on a real-world metaphor: Anything you represent on a screen if it is a metaphor for real-world entity then users would easily understand.
* Disclose information in a progressive fashion: The interface should be organized hierarchically i.e. on the main screen the information about the task, an object or some behavior should be presented first at a high level of abstraction. More detail should be presented after the user indicates interest with a mouse pick.

Make the interface consistent:

* Allow the user to put the current task into a meaningful context: Many interfaces have dozens of screens. So it is important to provide indicators consistently so that the user know about the doing work. The user should also know from which page has navigated to the current page and from the current page where can navigate.
* Maintain consistency across a family of applications: The development of some set of applications all should follow and implement the same design, rules so that consistency is maintained among applications.
* If past interactive models have created user expectations do not make changes unless there is a compelling reason.

5a.

5b. Tangling arises when one module implements (part of) several requirements; scattering arises where the implementation of a single requirement is spread across several modules.

2nd part

Say several requirements in a system have a shared component that implements the issuing of a confirmation to a user that some action has occurred. This is tangled system. If one of these changes so that some change to the confirmation is required, all of the other requirements that issue confirmations are affected by this.

If the implementation of each requirement which issues a confirmation is scattered across several modules, then all of these have to be checked that the change will not affect their operation.

5b. join point: An event in an executing program where the advice

associated with an aspect may be executed.

Pointcut: A statement, included in an aspect, that defines the join

points where the associated aspect advice should be

executed. In this example, the pointcut is a simple statement:

*before: call (public void update\* (..))*

WEB 2016/17

1A.differences btw html 5 and html

a. HTML5 supports SVG (Scalable Vector Graphics), canvas, and other virtual vector graphics, whereas in HTML,

using vector graphics was only possible by using it in conjunction with different technologies like Flash, VML (Vector Markup Language), or Silverlight.

b. Web SQL databases are used in HTML5 for storing data temporarily. Meanwhile, in the previous version of HTML, only browser cache could be utilized for this purpose.

c. With HTML5, JavaScript can run within a web browser, while the older HTML only allows JavaScript to run in the browser interface thread.

d. HTML5 is not based on SGML. This means that the language has improved parsing rules which provide enhanced compatibility.

You can use inline MathML and SVG

1. Video and Audio

Video and audio are the new tags which allow to embed a video in the website. YouTube also declare video embed by giving the code to embed for their videos.

It helps the web to be more involved with multimedia. A new tag is also available in HTML5 and that is audio tag. Which is used to embed any audio in the web.

<video controls preload>

    <source src="cohagenPhoneCall.ogv" type="video/ogg; codecs='vorbis, theora'" />

    <source src="cohagenPhoneCall.mp4" type="video/mp4; 'codecs='avc1.42E01E, mp4a.40.2'" />

    <p> Your browser is old. <a href="cohagenPhoneCall.mp4">Download this video instead.</a> </p>

</video>

<audio autoplay="autoplay" controls="controls">

 <source src="file.ogg" />

    <source src="file.mp3" />

    <a>Download this file.</a>

</audio>

2. nav

The nav element is used for the part of a internet site that links to different pages at the website. The hyperlinks can be organized a number of approaches.

below, the hyperlinks are displayed inside paragraph factors. An unordered list can also be used.

<nav>

  <a href="/html/">HTML</a>

  <a href="/css/">CSS</a>

  <a href="/js/">JavaScript</a>

  <a href="/jquery/">jQuery</a>

</nav>

3. header

The header element can be used to institution collectively introductory factors on a website,

such as a business enterprise brand, navigation objects, and occasionally, a search form.

<header>

  <img src="company-logo.png">

  <nav>

    <p><a href="login.html">Log In</a></p>

    <p><a href="signup.html">Sign Up</a></p>

    <p><a href="contact.html">Contact Us</a></p>

  </nav>

</header>

4. Figure and Figcaption

Earlier there was no way to of figure as well as give caption to that figure. But, with the introduction of figure as well as figcaption,

it has become semantically possible to insert an image in a page with its caption.

<figure>

    <img src="image/image-1.jpg" alt="About ADMEC" />

        <figcaption>

             <p>This is our institute </p>

        </figcaption>

</figure>

5. canvas

canvas is a tag of HTML which is newly introduced in HTML5. It is used to draw the images on the fly. It can be used for visual images, rendering graphs, game graphics.

Example of canvas

<canvas id="myCanvas"></canvas>

<script>

var canvas = document.getElementById("myCanvas");

var ctx = canvas.getContext("2d");

ctx.fillStyle = "#FF0000";

ctx.fillRect(0, 0, 80, 80);

</script>

1B.

Static Positioning

•HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.

•Static positioned elements are not affected by the top, bottom, left, and right properties.

div.static {

position: static;

border: 3px solid #73AD21;

}

Fixed Positioning

•An element with fixed position is positioned relative to the browser window.

•It will not move even if the window is scrolled:

Example

p.pos\_fixed

{

position:fixed;

top:30px;

right:5px;

}

Relative Positioning

•A relative positioned element is positioned relative to its normal position.

Example

h2.pos\_left

{

position:relative;

left:-20px;

}

h2.pos\_right

{

position:relative;

left:20px;

}

Absolute Positioning

•An absolute position element is positioned relative to the first parent element that has a position other than static.

If no such element is found, the containing block is <html>:

Example

h2

{

position:absolute;

left:100px;

top:150px;

2A. 7 HTTP requests

ii. <http://www.mysite.com/entrance/home.html>

<http://www.mysite.com/style.css>

<http://www.mysite.com/myscript.js>

<http://www.mysite.com/image/unilag_logo.png>

<http://www.mysite.com/image/vicmykid.jpg>

<http://www.mysite.com/mummy.jpg>

<http://www.mysite.com/image/mummy.jpg>

2B. The isset() function checks whether a variable is set, which means that it has to be declared and is not NULL.

This function returns true if the variable exists and is not NULL, otherwise it returns false.

<?php

$a = 0;

// True because $a is set

if (isset($a)) {

echo "Variable 'a' is set.<br>";

}

?>

The empty() function checks whether a variable is empty or not.

This function returns false if the variable exists and is not empty, otherwise it returns true.

<?php

$a = 0;

// True because $a is empty

if (empty($a)) {

echo "Variable 'a' is empty.<br>";

}

// True because $a is set

if (isset($a)) {

echo "Variable 'a' is set.";

}

?>

## 2Bii

* *1xx informational response* – the request was received, continuing process
* *2xx successful* – the request was successfully received, understood, and accepted
* *3xx redirection* – further action needs to be taken in order to complete the request
* *4xx client error* – the request contains bad syntax or cannot be fulfilled
* *5xx server error* – the server failed to fulfil an apparently valid request

|  |  |
| --- | --- |
| 100 Continue | The server has received the request headers, and the client should proceed to send the  request body |
| 101 Switching Protocols | The requester has asked the server to switch protocols |
| 103 Checkpoint | Used in the resumable requests proposal to resume aborted PUT or POST requests |
| 200 OK | The request is OK (this is the standard response for successful HTTP requests) |
| 201 Created | The request has been fulfilled, and a new resource is created |
| 202 Accepted | The request has been accepted for processing, but the processing has not been completed |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 300 Multiple Choices | A link list. The user can select a link and go to that location. Maximum five addresses. | | 301 Moved Permanently | The requested page has moved to a new URL | | 302 Found | The requested page has moved temporarily to a new URL | |  |

|  |  |
| --- | --- |
| 400 Bad Request | The request cannot be fulfilled due to bad syntax |
| 401 Unauthorized | The request was a legal request, but the server is refusing to respond to it. For use when  authentication is possible but has failed or not yet been provided |
| 402 Payment Required | *Reserved for future use* |
| 403 Forbidden | The request was a legal request, but the server is refusing to respond to it |
| 404 Not Found | The requested page could not be found but may be available again in the future. |

|  |  |
| --- | --- |
| 500 Internal Server Error | A generic error message, given when no more specific message is suitable |
| 501 Not Implemented | The server either does not recognize the request method, or it lacks the ability to fulfill the request |
| 502 Bad Gateway | The server was acting as a gateway or proxy and received an invalid response from the  upstream server |
| 503 Service Unavailable | The server is currently unavailable (overloaded or down) |
| 504 Gateway Timeout | The server was acting as a gateway or proxy and did not receive a timely response from the  upstream server |

3A. dbconnect.php file helps to connect our php application to the database

It takes in hostname, username, password, database name as input, it is then passed into a try and catch block and inside the try block a connection is created using PDO and the necessary parameters are passed in. then inside the catch block, if there is any PDO connection error, we output “Sorry, something went wrong”;

file\_put\_contents writes the error message into PDOErrors.txt

3B. citation.php first of all imports the dbconnect.php into citation.php file,

The next line checks if the submit function is a POST request and it is set.

Then set data from author into an author variable

Then set citation from POST data into a citation variable

Then set email into email variable.

Then a check is done to see if variable author and citation is not empty.

In the “if” block statement, we create a connection and use the insert query to store the data.

Then after creating connection with query, we execute the query.

Else if the variable author and citation are empty, we output “please enter a citation and the name of the author”

3c. <html>

<head>

<title>Form page</title>

</head>

<body>

<form method=”post” action=”citation.php”>

<input type=”email” name=”em” value=””>

<input type=”text” name=”citation” value=””>

<input type=”author” name=”em” value=””>

<button type=”submit” name=”Submit”>Submit</button>

</form>

</body>

</html>

3d. $requete = $conn->prepare(“INSERT INTO citation\_table(author,citation,email)VALUES($author,$citation,$email)”);

$requete->execute();

3e. [http://localhost/citation.php?email=”email\_value”&citation=”citation\_value](http://localhost/citation.php?email=)”&author=”author\_value”

4a.function changeDisplay(){

document.getElementById("text").style.color = "red";

document.getElementById("text").style.fontSize = "14px";

document.getElementById("text").style.fontfamily = "lato";

document.getElementById("text").value = document.getElementById("msg").value;

}

4.

2017/2018

1a. [www.unilag.edu.ng/student.php?id=123&name=mishael](http://www.unilag.edu.ng/student.php?id=123&name=mishael)

[www.unilag.edu.ng/student.php?matric=170805526&name=daniel](http://www.unilag.edu.ng/student.php?matric=170805526&name=daniel)

1b. The Web relies on three mechanisms to make information exchange and resource sharing possible:

•A uniform naming scheme for locating resources on the Web e.g. URIs

•Protocols for access to named resources over the Web e.g. HTTP

•Hypertext for easy navigation among resource e.g. HTML

1c. PHP session is used to store and pass information from one page to another temporarily (until user close the website).

PHP session technique is widely used in shopping websites where we need to store and pass cart information e.g. username, product code, product name, product price etc from one page to another.

PHP session creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.

You want the alternative to cookies on browsers that do not support cookies.

You want to store global variables in an efficient and more secure way compared to passing them in the URL.

Session can take other datatypes.

Session contain data about the current user. It can be used across pages.

They are accessible to all pages contained in a single web app.

2a.

The descendant selector matches all elements that are descendants of a specified element.

div p {

background-color: yellow;

}

The child selector selects all elements that are the children of a specified element.

div > p {

background-color: yellow;

}

2b.

An element with position: relative; is positioned relative to its normal position:

div.relative {

position: relative;

left: 30px;

border: 3px solid #73AD21;

}

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

div.absolute {

position: absolute;

top: 80px;

right: 0;

width: 200px;

height: 100px;

border: 3px solid #73AD21;

}

2c.

<html>

<table style="width:100%">

<tr>

<th>Matric</th>

<th>Name</th>

<th>Telephone</th>

<th>Email</th>

</tr>

<tr>

<td>080505014</td>

<td>Adeola Adetunji</td>

<td>08073221458</td>

<td>adeola@adetunji.com</td>

</tr>

<tr>

<td>090505080</td>

<td>Abdul Philips </td>

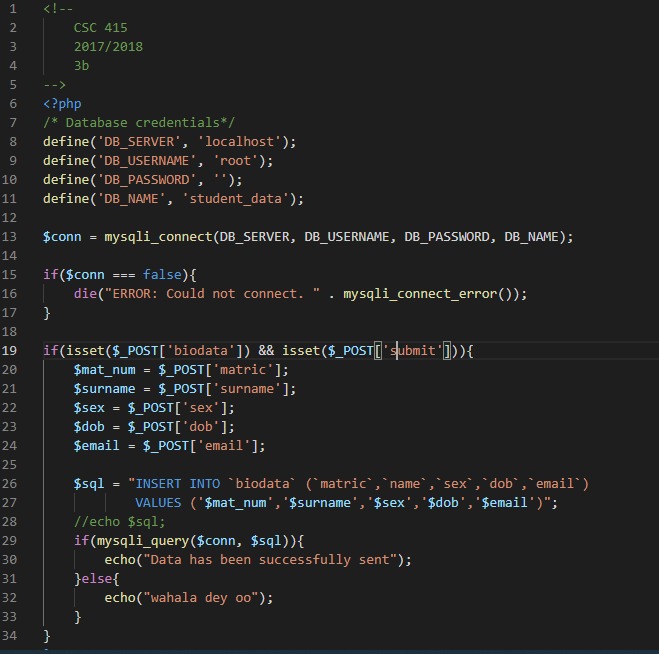
<td>017102053</td>

<td>abdul@philips.com</td>

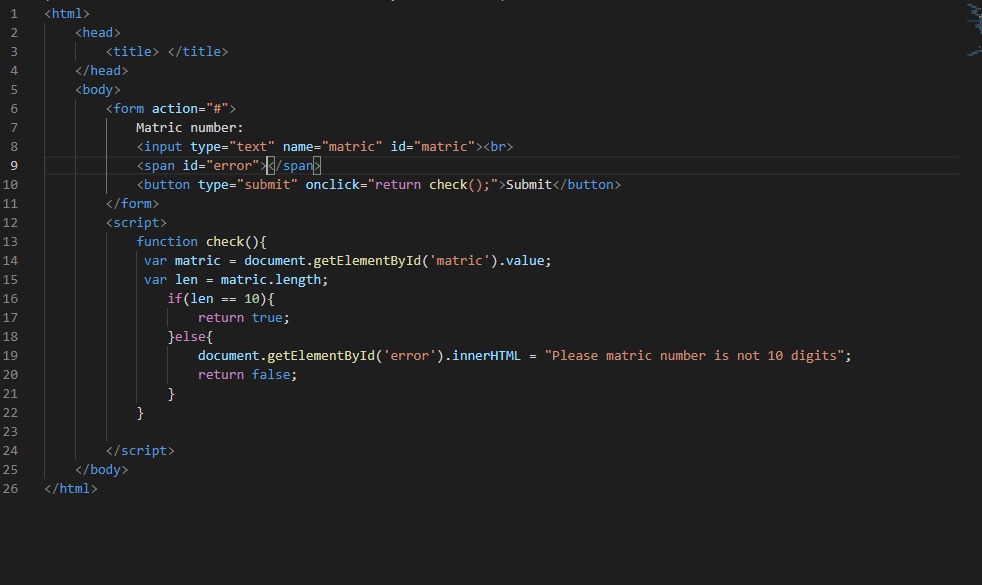
</tr>

</table>

</html>

3a. 3b.

http://www.exam.com/biodata.php?matricnumber=80929&name=isioma&sex=female&dateofbirth=2000/03/12&email=isiomaiyegbuniwejohn@gmail.com

3c. 

4a. foreach($services as $key => $value){

echo “Main = $key”;

{

foreach($value as $key => $v){

echo “{Key=$key, Value = $v} <br>”;

}

}

4b.

The cookie is a message that the web server pass to your web browser when you visit the internet sites , Your browser stores the message in a small file , When you request another page from the server , your browser sends the cookie back to the server .

The cookie is a small file stored on the user’s computer , It is used by the websites to track the visitors to keep the user information such as the username , interests , Password Remember Option , etc , If any web application use cookies , The server send cookies & client browser will store it .

The cookies are domain specific , The domain can not read or write to the cookie created by another domain ,

It is done by the browser for security purpose , The cookies are browser specific , Each browser stores the cookies in a different location ,

The cookies created in one browser ( e.g in Google Chrome ) will not be accessed by another browser ( Internet Explorer/Firefox ) .

What is cookie?

A cookie is a small piece of text file stored on user's computer in the form of name-value pair. Cookies are used by websites to keep track of visitors e.g. to keep user information like username etc. If any web application using cookies, Server send cookies and client browser will store it. The browser then returns the cookie to the server at the next time the page is requested.

The most common example of using a cookie is to store User information, User preferences, Password Remember Option etc

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Disadvantages of cookies

The cookies are not secure as they are stored in a clear text & no sensitive information should be stored in cookies ,

They may pose to a possible security risk because anyone can open & tamper with the cookies .

Advantages of using cookies

Here are some of the advantages of using cookies to store session state.

· Cookies are simple to use and implement.

· Occupies less memory, do not require any server resources and are stored on the user's computer so no extra burden on server.

· We can configure cookies to expire when the browser session ends (session cookies) or they can exist for a specified length of time on the client’s computer (persistent cookies).

· Cookies persist a much longer period of time than Session state.

Disadvantages of using cookies

Here are some of the disadvantages:

· As mentioned previously, cookies are not secure as they are stored in clear text they may pose a possible security risk as anyone can open and tamper with cookies. You can manually encrypt and decrypt cookies, but it requires extra coding and can affect application performance because of the time that is required for encryption and decryption

· Several limitations exist on the size of the cookie text(4kb in general), number of cookies(20 per site in general), etc.

· User has the option of disabling cookies on his computer from browser’s setting .

· Cookies will not work if the security level is set to high in the browser.

· Users can delete a cookies.

· Users browser can refuse cookies,so your code has to anticipate that possibility.

· Complex type of data not allowed (e.g. dataset etc). It allows only plain text (i.e. cookie allows only string content)

4ci. Userid – name of cookie,

Odumuyiwa – value

Time- expiry time

/ - path(where cookie will be valid)

NB: / - for all pages. No slash – not all pages.

2nd part.

<?php  
$cookie\_name = "user";  
$cookie\_value = "John Doe";  
setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/"); // 86400 = 1 day  
?>

4cii. setcookie(userid, “ ”, time() – 3600);

4ciii. $\_COOKIE is a global variable(superglobals) used to retrieve the value of the cookie “user”.

2nd part - $\_SERVER, $\_REQUEST, $\_ENV, $\_GET, $\_POST.

2015/2016

**1a. i.**

h1 {

color: #36CFFF;

}

**ii.**

\* {

color: #000000;

}

**iii.**

ul em{

color: #000000;

}

**iv.**

input[type="text"]{

color: #000000;

}

**v.**

body > p {

color: #000000;

}

**vi.**

.black {

color: #000000;

}

**vii.**

input[type="date"]{

color: #000000;

}

b. i. Inline styles are used for specific instances within tags in the HTML document



ii. Are contained within the document.

Reside in <head> portion of HTML document

<head><title>Internal Style>/title>

<style type="text/css">

body{

background-color: #FFCC33;

}

.style1{

color:#FFFFFF;

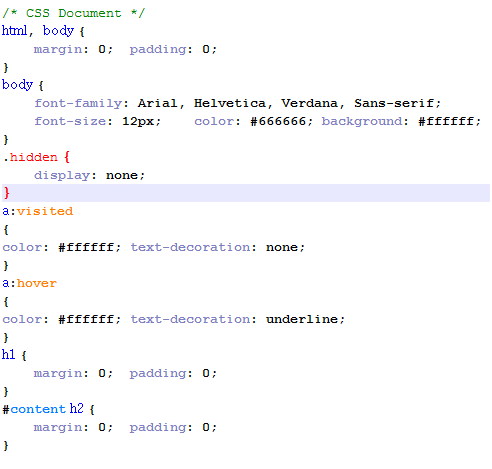
}

</style></head>

iii. Separate files containing CSS rules that are linked to pages in a website

<head><link rel="stylesheet" type="text/css”href="style.css" />

</head>



1c. a. HTML5 supports SVG (Scalable Vector Graphics), canvas, and other virtual vector graphics, whereas in HTML,

using vector graphics was only possible by using it in conjunction with different technologies like Flash, VML (Vector Markup Language), or Silverlight.

b. Web SQL databases are used in HTML5 for storing data temporarily. Meanwhile, in the previous version of HTML, only browser cache could be utilized for this purpose.

c. With HTML5, JavaScript can run within a web browser, while the older HTML only allows JavaScript to run in the browser interface thread.

2a.

* 2b. *1xx informational response* – the request was received, continuing process
* *2xx successful* – the request was successfully received, understood, and accepted
* *3xx redirection* – further action needs to be taken in order to complete the request
* *4xx client error* – the request contains bad syntax or cannot be fulfilled
* *5xx server error* – the server failed to fulfil an apparently valid request

|  |  |
| --- | --- |
| 100 Continue | The server has received the request headers, and the client should proceed to send the request body |
| 101 Switching Protocols | The requester has asked the server to switch protocols |
| 103 Checkpoint | Used in the resumable requests proposal to resume aborted PUT or POST requests |
| 200 OK | The request is OK (this is the standard response for successful HTTP requests) |
| 201 Created | The request has been fulfilled, and a new resource is created |
| 202 Accepted | The request has been accepted for processing, but the processing has not been completed |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 300 Multiple Choices | A link list. The user can select a link and go to that location. Maximum five addresses. | | 301 Moved Permanently | The requested page has moved to a new URL | | 302 Found | The requested page has moved temporarily to a new URL | |  |

|  |  |
| --- | --- |
| 400 Bad Request | The request cannot be fulfilled due to bad syntax |
| 401 Unauthorized | The request was a legal request, but the server is refusing to respond to it. For use when authentication is possible but has failed or not yet been provided |
| 402 Payment Required | *Reserved for future use* |
| 403 Forbidden | The request was a legal request, but the server is refusing to respond to it |
| 404 Not Found | The requested page could not be found but may be available again in the future. |

|  |  |
| --- | --- |
| 500 Internal Server Error | A generic error message, given when no more specific message is suitable |
| 501 Not Implemented | The server either does not recognize the request method, or it lacks the ability to fulfill the request |
| 502 Bad Gateway | The server was acting as a gateway or proxy and received an invalid response from the upstream server |

2c.

The *Host* request-header field is used to specify the Internet host and the port number of the resource being requested.

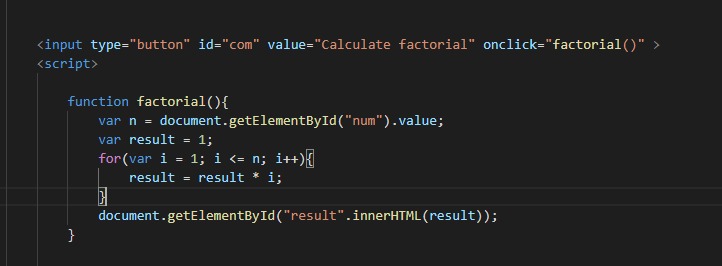
The *User-Agent* request-header field contains information about the user agent originating the request. Following is the general syntax:

The *Accept* request-header field can be used to specify certain media types which are acceptable for the response.

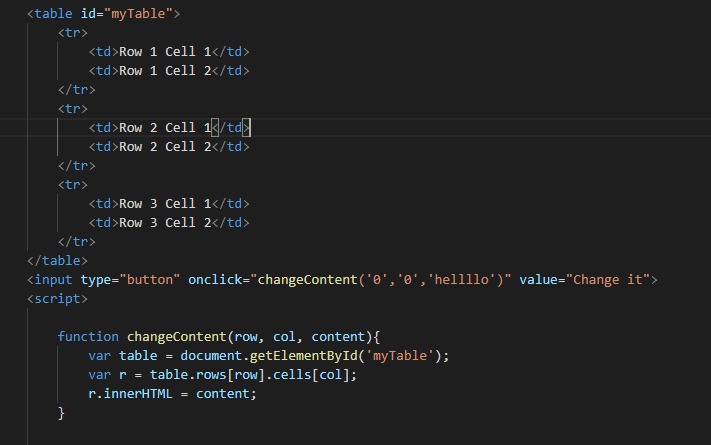
The *Accept-Language* request-header field is similar to Accept, but restricts the set of natural languages ‘that are preferred as a response to the request.

The *Accept-Encoding* request-header field is similar to Accept, but restricts the content-codings that are acceptable in the response.

The *Referer* request-header field allows the client to specify the address (URI) of the resource from which the URL has been requested.

3aiii. 

3b.

4b.

4c.

421/416

2018/19

* + - 1. B
      2. B
      3. C
      4. D
      5. B
      6. C
      7. A
      8. B
      9. A
      10. B
      11. A
      12. C
      13. A
      14. C
      15. B
      16. A
      17. C
      18. B
      19. D
      20. D

THEORY

1A. Invisibility: Software Projects progress are not visible to all

Complexity: per unit of cost spent, these are most complex.

Flexibility: subject to high degree of change.

IB. Individuals over processes

* + 1. working software over comprehensive doers.
    2. Customer collab over contract negotiations.
    3. Responding to change over following a .

IC. Project Manager skills are:

i. Effective Communiation

ii. Negotiation

iii. Leadership

iv.Coordination

v.Decision Making

vi.Motivation

vii. Motivation

viii. Coaching

ix. Facilitation

2A. 4 steps in Risk Mgt

Risk identification: Identify project, product and business risks(List of potential risks).

Risk analysis: assess the likelihood(probability) and consequences(impacts) of these risks. Prioritized risk list.

Risk planning: draw up plans to avoid or minimize the effects of the risk. Risk avoidance and contingency plans.

Risk monitoring: monitor the risks throughout the projects.(Risk assessment).

2B. Examples of Software Project Risks

----Estimation- time requirement to develop the software is underestimated, size of software is underestimated, rate of defect repair is underestimated.

----Organizational: financial problems force resuctions in the project budget.

-----People: key staff are ill and unavailable at critical times, required training for staff is not available.

-----Requirements

-----Technology

-----Tools

2C. Insufficient number of well-trained IT Workers.

Financial constraints.

Inadequate regulation to protect IP.

Technological limitation.

Frequent change of user requirement.

Inadequate exposure / knowledge of a few IT graduates.

Lack of proper Organizational structures.

Underestimation of Project Constraints, especially scope, cost and time for the software development.

3A. Forming: This phase is where the team meets and learns about the project and what their formal roles and responsibilities are. Team members tend to be independent and not as open in this phase.

Storming: During this phase, the team begins to address the project work, technical decisions, and the project management approach. If team members are not collaborative and open to differing ideas and perspectives the environment can become destructive.

Norming: In the norming phase, team members begin to work together and adjust work habits and behaviours that support the team. The team begins to trust each other.

Performing: Teams that reach the performing stage function as a well organized unit. They are interdependent and work through issues smoothly and effectively.

Adjourning: In the adjourning phase, the team completes the work and moves on from the project. The duration of a particular stage depends upon team dyanmics, team size, and team leadership. Project managers should have a good understanding of team dynamics in order to move their team members through all stages in an effective manner.

3B. Steps to ensure team success

* Establish the Team: Start with the Project Mgt Team.
* Facilitate Effective communication.
* Encourage collaboration.
* Accept and manage problems.
* Recognition and Reward.

3C. Storming stage.

2nd part: Talk to your team about the team development model.(accept that it is part of the team development process.

Clarify team goals and individual roles and responsibilities.

Speak to your team members one-on-one.

CSC 421

2017/2018

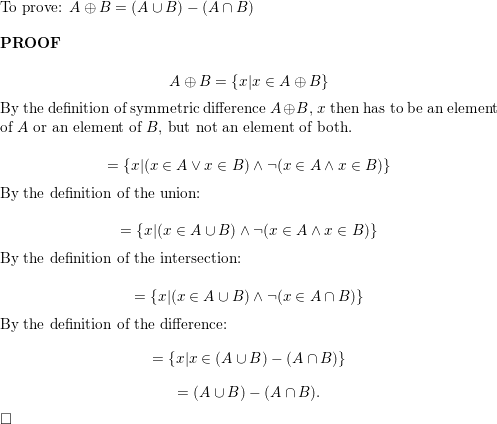
SECTION A

* + - 1. D
      2. B
      3. A
      4. B
      5. A
      6. A
      7. B
      8. B
      9. B
      10. C

DISCRETE MATHS

2017/18

1B.



1C.

Solution: The graphs G and H both have eight vertices and 10 edges. They also both have four

vertices of degree two and four of degree three. Because these invariants all agree, it is still

conceivable that these graphs are isomorphic.

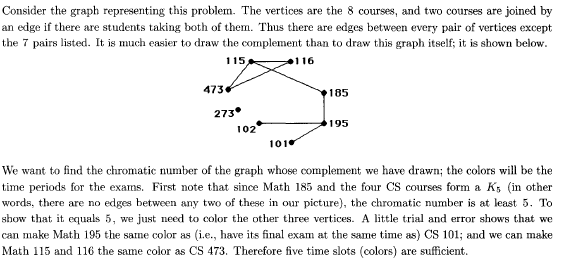
However, G and H are not isomorphic. To see this, note that because deg(a) = 2 in G, a

must correspond to either t , u, x, or y in H, because these are the vertices of degree two in H.

However, each of these four vertices in H is adjacent to another vertex of degree two in H,

which is not true for a in G.

2A.



2B. **NP Problem:**   
The NP problems set of problems whose solutions are hard to find but easy to verify and are solved by [Non-Deterministic Machine](https://www.geeksforgeeks.org/difference-between-deterministic-and-non-deterministic-algorithms/) in polynomial time.

[**NP-Hard Problem**](https://www.geeksforgeeks.org/tag/nphard/)**:**   
A Problem X is NP-Hard if there is an NP-Complete problem Y, such that Y is reducible to X in polynomial time. NP-Hard problems are as hard as NP-Complete problems. NP-Hard Problem need not be in NP class.

[**NP-Complete Problem**](https://www.geeksforgeeks.org/algorithms-gq/np-complete-gq/)**:**

A problem X is NP-Complete if there is an NP problem Y, such that Y is reducible to X in polynomial time. NP-Complete problems are as hard as NP problems. A problem is NP-Complete if it is a part of both NP and NP-Hard Problem. A non-deterministic Turing machine can solve NP-Complete problem in polynomial time.

2Bii. Traveling salesman

Knapsack

Graph Coloring.

3A.

Proof: Show that all of the properties of an equivalence relation hold.  Reflexivity: Because l(a) = l(a), it follows that aRa for all strings  a.   Symmetry: Suppose that aRb.  Since l(a) = l(b), l(b) = l(a) also  holds  and therefore bRa.   Transitivity: Suppose that aRb and bRc. Since l(a) = l(b),and  l(b) = l(c), l(a) = l(c) also holds and therefore aRc. (QED)

OR

Since length(a) = length(a), we can say a R a and therefore, R is reflexive. Suppose a R b, so length(a) = length(b). Then, length(b) = length(a) so R is also symmetric. Finally, suppose a R b and b R c; this means length(a) = length(b) and length(b) = length(c). We see that length(a) = length(c) so R is also transitive and is thus an equivalence relation.

3B. gcd(75,12)

75= 12 \* 6 +3

Gcd(12,3) = 3

3C. 20C2 + 20C1 + 20C0 = 211 = 2.01 \* 10-4 = 0.00020

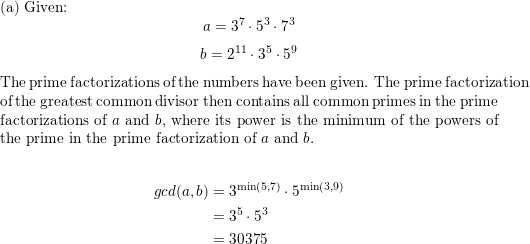
220 1048576

4a.

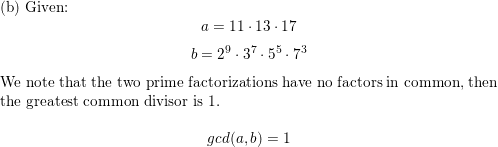
The pair (1, 1) is in R1, R3, R4, and R6; (1, 2) is in R1 and R6; (2, 1) is in R2, R5,

and R6; (1,−1) is in R2, R3, and R6; and finally, (2, 2) is in R1, R3, and R4.

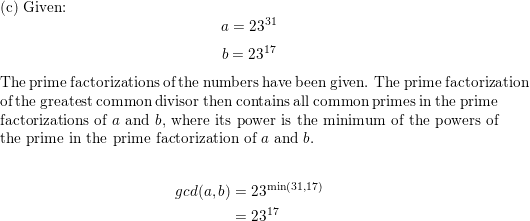
4b. **1**



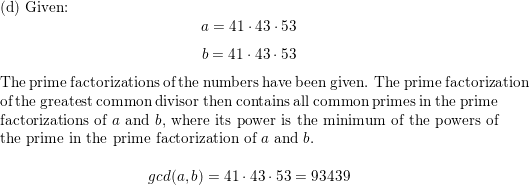
**2**



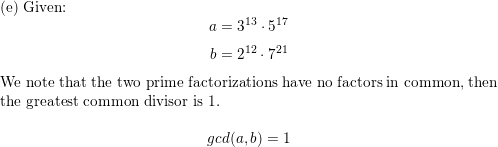
**3**



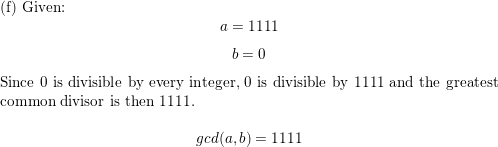
**4**

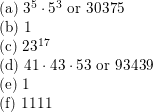


**5**

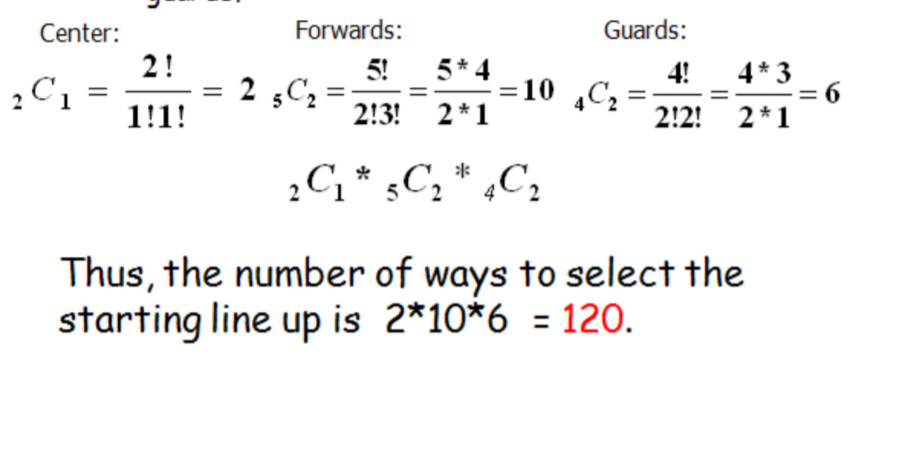


**6**

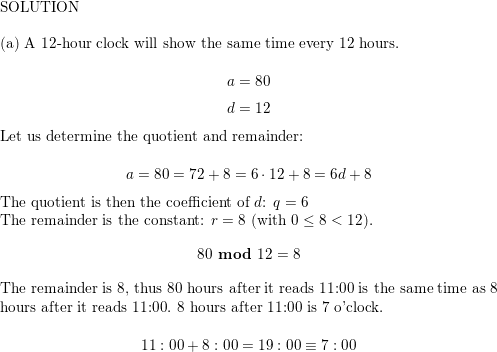


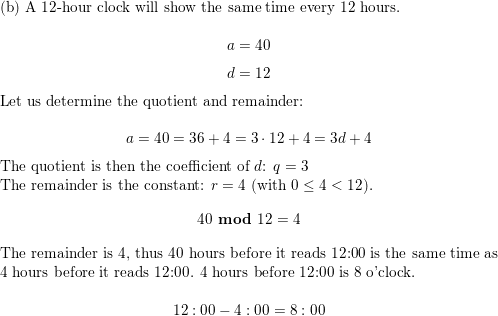


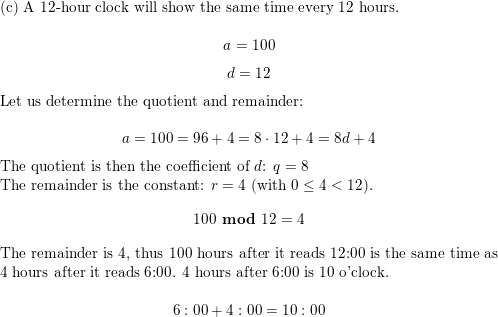
4c.



5b.







5ci.

10! = 2.3.4.5.6.7.8.9.10

2. 3. 22 .5 . (2.3) . 7 .2 . 23 . 32 . (2.5)

28 .34 .52 . 7

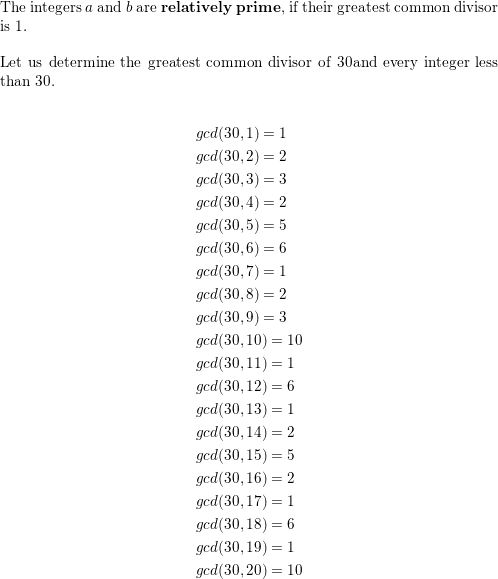
EXPLANATION

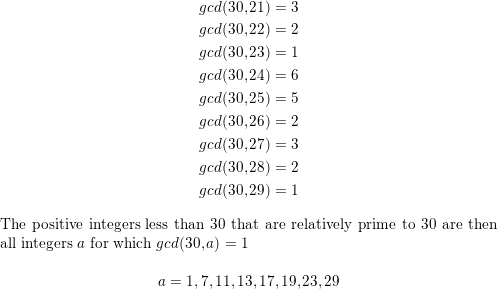
First compute two’s power.  There are five numbers between one and ten that two divides into.  These numbers are given 2\*1, 2\*2, …, 2\*5.  Further, two also divides two numbers in the set {1,2,3,4,5}.  These numbers are 2\*1 and 2\*2.  Continuing in this pattern, there is one number between one and two that two divides into.  Then a=5+2+1=8.

Now look at finding three’s power.  There are three numbers from one to ten that three divides into, and then one number between one and three that three divides into.  Thus b=3+1=4.  In a similar fashion c=2.  Then the set R={8,4,2,1}.  The final answer is:

10!=2^8*3^4*5^2*7

ii.





## 5ciii.

## A student can chose 10 questions from the given 13 questions by

### (i)      4 from part A and 6 from part B

### (ii)    5 from part A and 5 from part B

### (iii)    6 from part A and 4 from part B

### So, total ways = https://images.topperlearning.com/topper/questions/9633_image008.gif

### https://images.topperlearning.com/topper/questions/9633_image009.gif