

Sessional Marks Detail:

<https://docs.google.com/spreadsheets/d/1Uztx8nR0w09Ig0H76-u-709zql6loP-DZp7HCgd3vW8/edit?usp=sharing>

Assignments Repositories

https://docs.google.com/spreadsheets/d/1gN8m01-J2N_TBaY57tY6KzepdTQrmCHrMH3JjF23bj0/edit#gid=0

Classes

CS: Monday/Wednesday(8:15 am) - Room # 3

SE: Monday/Wednesday(9:45 am) - Room # 13

Lab

CS: Monday (1:30 PM - 4:30 PM) - Lab A or Main Lab

SE: Monday (1:30 PM - 4:30 PM) - Main Lab

Instructor: Bilal Shahzad (bilal.shahzad@pucit.edu.pk)

TAs:

CS TA: (IFTIKHAR LIAQUAT) bsef14m034@pucit.edu.pk

SE TA: SANA KANWAL BCSF14M019@pucit.edu.pk

To access course helping videos

<http://learninginurdu.pk/>

<https://www.youtube.com/channel/UCh1JI7ASy1EnzaM0hdVcuAQ/playlists>

<https://www.facebook.com/LearningInUrduCentre/>

<https://github.com/bilalshahzad139>

Assignments Calendar

Assign #	Description	Prerequisite	Deadline
1	Security Management System https://docs.google.com/document/d/1h/CzuSi6f9QuF_2faD1pYIEeHIB4Wfna_uZIUOpPPac/edit	Understanding of CSS, HTML, JavaScript, bitbucket, git	11-March-2018
2	Security Management System v2 https://docs.google.com/document/d/1bdWK7W2DQswO71RDVv-_PbwSMDnjyuQnAwBU7B6CDts/edit	PHP, MySQL	25-March-2018
3	Security Management System v3 https://docs.google.com/document/d/1-4hohI9T_WZztOKEMLKTLzfIRm-x1sljksEUN1rpLcg/edit?usp=sharing	PHP Api, jQuery, AJAX	03-April-2018
4	Graphical User Authentication	C#, Windows App Creation	15-April-2018
	After Mid Term		
6			W2 end
7			W3 end
8			W5 end
9			
10			

Quiz Calendar

Quiz # 1	14-March-2018	HTML, CSS, JavaScript (Lecture 1-3)
Quiz # 2	28-March-2018	PHP (Lecture 4-6)
Quiz # 3	04-April-2018	jQuery (Lecture 7-9)

Labs Detail

Lab 1	05-Mar-18	JavaScript Practice Exercises https://github.com/bilalshahzad139/JavaScript-Tasks-For-Practice
Lab 2	12-Mar-18	PHP Practice Exercises https://github.com/bilalshahzad139/PHP-Tasks-For-Practice
Lab 3	19-Mar-18	jQuery Practice Exercises https://github.com/bilalshahzad139/jQuery-Tasks-For-Practice
Lab 4	26-Mar-18	1) From PHP Practice Tasks https://github.com/bilalshahzad139/PHP-Tasks-For-Practice a) Task 14 b) Task 15 c) Task 16 2) From Learn jQuery Tasks https://github.com/bilalshahzad139/Learn_jQuery a) Dynamic Rows Generation b) ModalPopup c) Editable d) FormPrac
Lab 5	02-Apr-18	Win Form Practice
Lab 6	09-Apr-18	ADO.NET Practice

Software Required to be installed

- XAMPP
- Notepad++/Sublime/Any text editor you are comfortable with
- MS Visual Studio 2015
- MS SQL Server Express Edition
- Fiddler

Lecture 1 (26-Feb-2018) CSF15 + SEF15

- Introduction to Course
- Evaluation
 - Quiz
 - Test
 - Assignments
 - Class Participation
 - Mid Term
 - Final Term
- Introduction to HTML
 - Tag
 - Attribute
 - Block element vs. Non-Block Element
 - Elements who have any impact in renderings vs. elements who doesn't have
- Client Server Architecture
 - Thin Client - Thick Server
 - Thick Client - Thin Server
- Client side examples
 - Browser
 - Mobile App
 - Desktop App
- Examples of Web Servers
 - Apache
 - IIS (Internet Information Services)
 - Tomcat
- Examples of Server side environments/programming languages
 - PHP
 - JSP (Java Server Pages)
 - ASP (Active Server Pages)
 - ASP.NET
 - NodeJS (JavaScript server side environment)
 - Python
 - RoR
- Browser Main components
 - HTML Renderer
 - CSS Engine
 - JavaScript Engine
- HTTP (Hypertext Transfer Protocol)
 - Request Response Protocol
 - Request goes to Server, Processed there and Response is returned to the client (browser for example)
 - HTTP Port => 80

- HTTPS Port => 443
- Introduction to CSS (Cascading Style Sheet)
- How CSS can be written
 - Inline
 - At Page level using <Style> tag
 - In separate .css file
- CSS Rules/Selectors
 - Tag based
 - Class based
 - ID based
 - Attribute based
 - Mixture of above
 - Plus many other rules exist
- CSS BOX Model
 - Border
 - Padding
 - Margin
 - Content
- Positioning
 - Static
 - Fixed
 - Relative
 - Absolute

Helping CSS Videos

<http://learninginurdu.pk/category/css/>

You may face problem while setting Padding or Margin with non-block elements, check this post + answer to understand the issue and solution

<https://stackoverflow.com/questions/14604541/padding-top-not-working>

Lecture 2 (28-Feb-2018) CSF15 + SEF15

- Introduction to JavaScript
 - Inline
 - Internal (using <script> tag)
 - External (in a file with .js extension)
- Client side programming language
- Server side version is called NodeJS
- “var” keyword
- Global variable (global object is “window” in browser)
- Data Types
 - Number
 - Boolean
 - Undefined
 - Null
 - String
 - Date
 - Array
 - Object
- Conditional Statements & Loops
 - If, if else
 - Switch
 - For loop
 - While, do while
- functions
- Arrays
 - Associative Arrays (Key based)
- For-in loop
- Object
 - Is just like associative array
- XML format, JSON format
- DOM Manipulation using JavaScript

Practice Exercises (Lab 1)

<https://github.com/bilalshahzad139/JavaScript-Tasks-For-Practice>

Helping Video Material

<http://learninginurdu.pk/category/javascript/>

For your self learning (optional from course point of view):

Go through all these files, run pages, read comments given in code.

https://github.com/bilalshahzad139/Learn_JavaScript

Lecture 3 (05-March-2018) CSF15 + SEF15

- Pending content from Lecture 2 for CS
 - Understanding of Functions in JavaScript
 - Anonymous Function
 - Named Function
 - When you create a function in JavaScript it is of type "Function". You may use constructor approach to create a function. For example
 - `Var f = new Function(a,b,"return a+b");`
 - In above example we have created a function which is taking two parameters (a & b) and last parameter is body of our function. Variable "f" is pointing to newly created function object.
 - So if function is also like an object, you may hold its reference in variable. It means you may add it in array, you may send it to another function as argument etc.
 - In a function **this** refers to the caller. It is similar concept like we've in other languages.
 - Understanding of Objects in JavaScript
 - In JavaScript there is a reserved **class** keyword but it does nothing. It means you can't create classes in JS like other OO languages.
 - But JavaScript is OOP language. You use different tricks to achieve different behaviors of OO.
 - In JS, top level type name is "Object". You can create an instance of it by using two approaches 1) direct/literal approach 2) constructor approach
 - Constructor Approach: `var obj = new Object();`
 - Literal(direct approach): `var obj = {};`
 - Above code will create an empty object and obj will be pointing to it. Now you may add new properties (data member & functions) in it by using 1) dot operator 2) key based syntax. In JS, objects are like associative arrays.
 - `obj.ID = 1;` is similar to `obj["ID"] = 1;`
 - `obj.Name = "ABC"` is similar to `obj["Name"] = "ABC";`
 - When you set a value to a property in JS, it see if there is any such property in object, if there is set value to that if not create a new property and then set.
 - If you will try to get value from a property which is not part of object, you will get **undefined** as value.
 - When you create object using 'literal' notation, you may also initial object with properties
 - `var obj = {ID: 1, Name:"ABC",Age: 20};`
 - We also call this JavaScript Object Notation (JSON)
 - And when we represent this in string format, we call it JSON string.

- So let's say you have a JS (created by whatever approach), you can get JSON string from it using **JSON.stringify(obj)** function. And if you have JSON string, you can create JS object again by using `JSON.parse(str)`;
 - For example
 - `var obj = new Object();`
 - `obj.ID = 10; obj.Age = 20;`
 - `var str = JSON.stringify(obj);`
 - str will contain **"{ID:10,Age:20}"** please note that this is string.
 - `var o = JSON.parse(str);`
 - Here **o** is a variable which is pointing to an object. This object contains ID & Age as properties. You may access them using `o.ID` or `o.Age` for example.
- JSON format is very popular now a days for data representation. Many languages provide support to convert their objects into JSON format or convert JSON formatted string back to their object.
- XML is also a format and had been very popular in past for same purpose (representing data)
- JavaScript in Browser
 - JavaScript is Used as Client side programming language in Browser
 - **window** is root object in browser
 - There is a type (constructor) available for every html element in browser For example
 - For `<html>` there is `HTMLHtmlElement`
 - For `<div>` there is `HTMLDivElement`
 - When browser parses your html, it generates similar object tree using corresponding classes. This tree is call DOM (Document Object Model)
 - **document** object is used to play with DOM.
 - `document.getElementById()`
 - `document.createElement()`
 - Normally JS was used in browser mainly for data validation (client side validation) but for last decade client side development is increased. Now clients are becoming "Thick Clients". You get data from server using JS and then generates your HTML on client side.
- Introduction to Client-Server Communication
- Introduction to HTTP
- How Processing happens on web server
 - Static Content: This is served as it is. Server knows which types of resources are to be considered as static resources. For example, HTML pages, JS, CSS, JPG, BMP, EXE.

- Dynamic Content: Some processing is done and then response is generated. Server knows which types of resources to be considered as dynamic resources. For example .PHP, .ASP, .ASPX, .JPG, .WCF
- Web Server prepares response (header + body). Header contains meta information(response status, server detail etc.). Body contains actual content (e.g. html).
- What happens on client side
 - Once response (e.g. HTML) reaches on client side, browser parses <head> section and load all JS/CSS files mentioned in <head> section. Once head is loaded, it starts parsing <body> section. During parsing of body, if it finds any static resource declaration (e.g. <img, <script), it sends a separate HTTP request to download that content. Once whole <body> is parsed and DOM is created, DOMReady event is fired. And when all external resources (images, css, js etc.) are downloaded, Body Load event is fired.
- Introduction to PHP
- Variable Declaration
- <?php tag
-

Lecture 4 (07-March-2018) CSF15 + SEF15**Task 1:**

- Creating a new folder in htdocs and then a file "test.php"
- Write basic html and then <?php ?> block
- Then writing some echo statements for testing
- Then writing multiple <?php ?> blocks in page
- Understanding the difference of HTML shown in page than actual code file

Task 2:

- Rendering Hello Message with counter using Loop
- Rendering different textboxes and showing counter as value using Loop

Task 3:

- Creating sample login form (by providing "name" to each element)
- Understanding what happens when Login or Registration button is clicked
- Understanding what happens when no method or GET method or POST method is mentioned in form
- Understanding what is role of "action" attribute in form tag

Task 4:

- Creating a new abc.php file
- Understanding how to get value from HTTP request packet using \$_GET or \$_POST or \$_REQUEST
- Understanding how to check if a value exists or null using **isset** function
- Hard coding Login=admin & Password=admin in code

Task 5:

- Quick review of some sample PHP files (which are being used in videos)
- Understanding of Arrays

Helping Video Material

<http://learninginurdu.pk/category/php/>

Practice Exercises (Lab 2 - 12-March-2018)

<https://github.com/bilalshahzad139/PHP-Tasks-For-Practice>

Lecture 5 (12-March-2018) CSF15 + SEF15

- Discussion on Arrays
- Discussion on Controlling HTML output using PHP

Task 6:

- Getting posted data in same page
- Authenticate user
- If user is authenticated, redirect him to home page
- If user is not authenticated, show him error message and also login should be loaded with entered value
- Apply Server side validation

Task 7:

- Displaying countries in dropdown using an array of countries
- Submitting and Showing selected country in output

Task 8:

- Quick Understanding of Server side Sessions
- Understanding of Session Variable
- Using Session for authorization (e.g. on home screen)

Quiz # 1 (14-March-2018)

Content: First 3 Lectures (HTML, CSS, JavaScript)

Practice Exercises (Lab 2 - 12-March-2018)

<https://github.com/bilalshahzad139/PHP-Tasks-For-Practice>

Helping Video Material

<http://learninginurdu.pk/category/php/>

PHP Files (used in Videos)

<https://github.com/bilalshahzad139/php-learn-in-urdu>

Lecture 6 (14-March-2018) CSF15 + SEF15

Task 9:

- Store some value in session
- Get value from Session
- Use Session data to authorize user on a page

Task 10:

- Creating a separate file which will contain some functions
- include/require that file in your page

Task 11:

- Creating a table in MySQL
- Adding dummy data in that table
- Creating connection
- Query & show data on page in tabular format

Task 12:

- Authentication from Database

Task 13:

- Saving data in database

- Run Video 9
- Check Video 10
- Create DB and a table countries
- Task17
- Task18
- Task19
- Task20
- Check Video 13

Quiz # 2 (21-March-2017)

Content: **PHP**

Lecture 7 (19-March-2018) CSF15 + SEF15

- Introduction to jQuery
 - JQuery.com
 - Minified vs. Non-Minified
 - DomReady vs BodyLoad events
 - Example of HTTP requests count
- jQuery Object
 - It encapsulates DOM objects inside it
 - If no element is found, still you have jQuery object with length zero
- jQuery vs. \$
- Using CSS Selectors to FIND elements from DOM
- Then ACT on jQuery Object e.g. .hide(), .show(), .closest(), .find()
- Pseudo selectors
 - :first, :last, :selected, :visible, :nth()

Task 1 Practice

Task 2 Practice

Task 2 Practice

jQuery Practice Exercises (Lab-3 work)

<https://github.com/bilalshahzad139/jQuery-Tasks-For-Practice>

jQuery Video Series for detailed learning

<http://learninginurdu.pk/category/jquery/>

More jQuery Practice exercises (For your own learning)

https://github.com/bilalshahzad139/Learn_jQuery

Quiz # 2 (21-March-2017)

Content: PHP

Lecture 8 (26-March-2018) CSF15 + SEF15

- Discussion on Assignment 2 & Assignment 3
- Synchronous vs. Asynchronous
- Introduction to AJAX (Asynchronous JavaScript and XML)
 - Create HTTP requests programmatically using JavaScript

https://github.com/bilalshahzad139/Learn_jQuery

Task 1:

Dynamic Rows Generation

Task 2:

Modalpopup

Task 3:

Editable

- Ajax Introduction
- API Introduction
- API Example

Lab 4 (26-March-2018)

1) From PHP Practice Tasks

<https://github.com/bilalshahzad139/PHP-Tasks-For-Practice>

a) Task 14

b) Task 15

c) Task 16

2) From Learn_jQuery Tasks

https://github.com/bilalshahzad139/Learn_jQuery

a) Dynamic Rows Generation

b) ModalPopup

c) Editable

d) FormPrac

Lecture 9 (28-March-2018) CSF15 + SEF15

Quiz # 2 is done

Task 4:

More Practice on AJAX

Debugging API response

Introduction to jQuery Plugins

- JQueryui.com

For Self Study (Part of course)

Web Basic Concepts

<http://learninginurdu.pk/category/web-concepts/>

Web Paths

<http://bstechnical.blogspot.com/2016/12/information-about-web-paths.html>

(Part 1 of EAD Course - Content Covered till Lecture # 9)

<http://learninginurdu.pk/category/html/>

<http://learninginurdu.pk/category/css/>

<http://learninginurdu.pk/category/javascript/>

<http://learninginurdu.pk/category/jquery/>

<http://learninginurdu.pk/category/php/>

<http://learninginurdu.pk/category/web-concepts/>

<http://bstechnical.blogspot.com/2016/12/information-about-web-paths.html>

Lecture 10 (02-Apr-2018)

- Microsoft Development Pre-2000 Era
 - Visual Studio (1-6)
 - Visual Basic 1-6
 - ASP (uses Visual Basic as programming Language)
- Microsoft Development Post-2000 era
 - .NET Framework (1.0, 2.0, 3.0, 3.5, 4.0, 4.5, 4.6.1, 4.6.2, 4.7.1)
 - .NET Core (1.0, 2.0) => Multi Platform supported
 - Visual Studio.NET
 - Visual Studio 2001, 2003, 2005, 2008, 2010, 2012, 2013, 2015, 2017
- MSIL (Microsoft Intermediate Language)
- CLR (Common Language Runtime)
- When a "Project" is compiled in .NET, it produces an "assembly" file
- Assembly: It contains MSIL + metadata of assembly
 - Assembly can be DLL (Dynamic Link Library), without Main method
 - Assembly can be EXE (Executable), with Main or Entry Point
- A Solution File is a text (XML) based file which contains information of projects
- A project file (.csproj for example) is a text (XML) based file which contains information about items of that project.
- One solution may have many projects
- A namespace is a logical partitioning of types
- We learnt
 - How to create a new Console Application
 - How to add a class library project in existing solution
 - How to add a windows form project in existing solution
 - How to add reference of a DLL/Project (from same solution)
 - What is use of "using" statement at top
 - Reading from & writing to Console
 - Usage of Convert class
 - Debugging in Visual Studio
 - Setting a project as startup project
 - Setting properties (e.g. Text) of form elements using "Properties" window
 - Button click event handling
 - Two number calculator
 - Solution Explorer, Output window, Watch window, Toolbox, Properties Window

Lab 5 (02-April-2018)

Task 1

1- Creating a new Windows Form based Project

- **Solution Name should be your roll number.**

- Creating a **Login screen** with Login & Exit Button
- Login button will call a method (ValidateUser) from BAL project and will redirect user to **Home** screen. Home screen will display the login (e.g. Welcome Admin). Hint: You may pass login (or any data) through constructor of Home screen.
- Password field should be password field (show display * when user will type)
- If Login or Password field is empty, Show alert to the user. Hint: Use MessageBox.Show() function to show alert.
- There will be logout button on Home screen. When user will click on it, hide current form and show Login form again.
- There will be one more button on home screen (GPA Calculator) which will open Task 2 screen.

2- Add a new class Library project (e.g. BAL)

- Create a class User which will have ValidateUser(login,password) method. It will do hard coding and will return true if provided login & password are "admin".

Task 2:

- GPA Calculator: Design a form which will contain 6 rows (one row for one subject). Each row will contains textboxes for
 - A checkbox
 - Total Marks:
 - Obtained Marks:
 - Credit Hours
- Provide a button "Calculate GPA". When user will click this button, you need to calculate GPA using the rows (where checkbox is checked). Also apply some validations on fields.

Help:

How to show another form:

- Create instance of that form and then call Show method with it.
 - MyForm frm = new MyForm()
 - frm.Show();

How to Hide current form

- this.hide();

How to exit application

- Application.Exit();