University of Colombo School of Computing
SCS 2208 - Rapid Application Development

Practical sheet 02

Javascript Practical

Conditional Statements (if else, switch), Loops (while, do while, for), Functions, Array Methods, OOP Concepts and JS Events

Exercises:

1. Write a solution for the below scenario using the if else condition.
   If the time is less than 10:00, create a "Good morning" greeting, if not, but the time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening".

2. Create a switch statement that will alert "Yellow" if the fruit is "banana", “Orange” if the fruit is "pineapple" and "Green" if the fruit is "apple".
3. Write a JS program to input three numbers and find the largest number.
4. Write a simple program to find whether the given number is prime or not.
5. Write a simple JS program to print odd numbers between 1 to 30.
6. Write a function that returns the square of the input number passed as an argument in the function call.
7. Write a function to reverse a given number.
   Sample input: 32243
   Sample output: 34223

8. Imagine you are developing a web-based library management system using JavaScript. One of the features you need to implement is a book borrowing system where users can borrow and return books. Design a JavaScript class `Library` that manages the inventory of books using an array. Each book is represented as an object with properties `title`, `author`, and `borrowed` (a boolean indicating if the book is currently borrowed).
   1. Implement methods within the `Library` class to perform the following operations:
      a. `addBook(title, author)`: Adds a new book to the library with the given title and author. The book should initially be marked as not borrowed.
      b. `findBook(title)`: Returns the index of a book in the library array based on its title. If the book is not found, return -1.

    c. `borrowBook(title)`: Marks a book as borrowed (if available) by updating its `borrowed` status to true.
    d. `returnBook(title)`: Marks a borrowed book as returned by updating its `borrowed` status to false.

2. Ensure that the `Library` class uses OOP concepts such as encapsulation (private array for books), inheritance (if applicable), and methods that operate on book objects.

3. Create a basic HTML interface with buttons or links to interact with the `Library` class. For example, a button to borrow a book (`onclick` event triggers `borrowBook(title)` method) and another to return a book (`onclick` event triggers `returnBook(title)` method).