Class 9 PHP Assignment

1. **Product Listing (Basic)**
   - You have an array of product objects, each with properties `name`, `price`, and `rating`. Write a function to sort the array by price in ascending order.
2. **Product Listing (Advanced)**
   - Extend the previous function to sort by multiple criteria: first by rating in descending order, then by price in ascending order if ratings are equal.
3. **Student Grades (Basic)**
   - You have an array of student objects, each with properties `name` and `grades` (an array of numbers). Write a function to sort the students by their average grade in descending order.
4. **Student Grades (Advanced)**
   - Extend the function to handle cases where students have the same average grade by sorting them alphabetically by name.
5. **Employee Records (Basic)**
   - You have an array of employee objects, each with properties `name`, `department`, and `salary`. Write a function to sort the array by department alphabetically.
6. **Employee Records (Advanced)**
   - Modify the function to then sort by salary within each department in descending order.
7. **Event Sorting**
   - You have an array of event objects, each with properties `title`, `date`, and `location`. Write a function to sort the array by date in ascending order.
8. **Library Books**
   - You have an array of book objects, each with properties `title`, `author`, and `publicationYear`. Write a function to sort the books first by author

alphabetically, then by publication year in descending order.

Regular Expressions

1. **Email Validation (Basic)**
   - Write a function that takes an email address as input and uses a regular expression to validate whether it is in a proper format.
2. **Email Validation (Advanced)**
   - Extend the function to check for common errors like consecutive dots in the local part or invalid characters.
3. **Phone Number Formatting (Basic)**
   - Write a function that takes a phone number as input and uses a regular expression to format it in a standard way (e.g., (123) 456-7890).
4. **Phone Number Formatting (Advanced)**
   - Extend the function to handle different formats of input (e.g., 1234567890, 123-456-7890, 123.456.7890) and convert them to the standard format.

forEach Loop on Associative Arrays

1. **User Permissions (Basic)**
   - You have an associative array (object) representing user permissions, where keys are user IDs and values are arrays of permission strings. Use a `forEach` loop to print each user ID and their permissions.
2. **User Permissions (Advanced)**
   - Modify the function to count how many users have a specific permission (e.g., 'admin') and log the count.
3. **Shopping Cart (Basic)**
   - You have an associative array (object) representing a shopping cart, where keys are product IDs and values are quantities. Use a `forEach` loop to calculate the total number of items in the cart.
4. **Shopping Cart (Advanced)**
   - Modify the function to also calculate the total cost of the items in the cart, given an array of product objects with `id`, `name`, and `price`

properties.

Forms - Handling Text Inputs

1. **Form Handling - Text Inputs (Basic)**
   - Create a simple HTML form with text inputs for `firstName`, `lastName`, and `email`. Write a function to capture the form data and print it to the console when the form is submitted.
2. **Form Handling - Text Inputs (Advanced)**
   - Extend the function to validate that none of the fields are empty and that the email is in a valid format. Display appropriate error messages for invalid inputs.
3. **Feedback Form (Basic)**
   - Create an HTML form with text inputs for `name`, `email`, and a textarea for feedback. Write a function to capture the form data and print it to the console when the form is submitted.
4. **Feedback Form (Advanced)**
   - Extend the function to validate that the `name` and `feedback` fields are not empty and that the email is in a valid format. Display appropriate error messages for invalid inputs.
5. **Survey Form (Basic)**
   - Create an HTML form with text inputs for `age`, `gender`, and a dropdown for `country`. Write a function to capture the form data and print it to the console when the form is submitted.
6. **Survey Form (Advanced)**
   - Extend the function to validate that the `age` field contains a valid number and the `gender` field is not empty. Display appropriate error messages for invalid inputs.
7. **Contact Form (Basic)**
   - Create an HTML form with text inputs for `phone` and `address`. Write a function to capture the form data and print it to the console when the form is submitted.
8. **Contact Form (Advanced)**

- Extend the function to validate that the `phone` field contains a valid phone number format and the `address` field is not empty. Display appropriate error messages for invalid inputs.

9. **Registration Form (Basic)**
    - Create an HTML form with text inputs for `username`, `password`, and `confirmPassword`. Write a function to capture the form data and print it to the console when the form is submitted.
10. **Registration Form (Advanced)**
    - Extend the function to validate that the `password` and `confirmPassword` fields match and that the `password` meets certain criteria (e.g., at least 8 characters). Display appropriate error messages for invalid inputs.