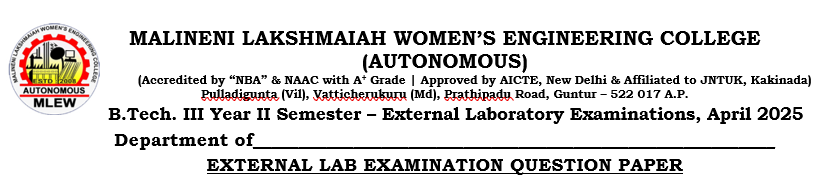
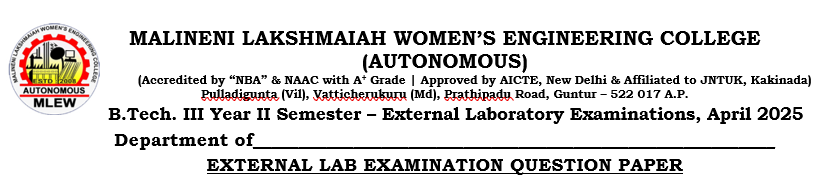
****

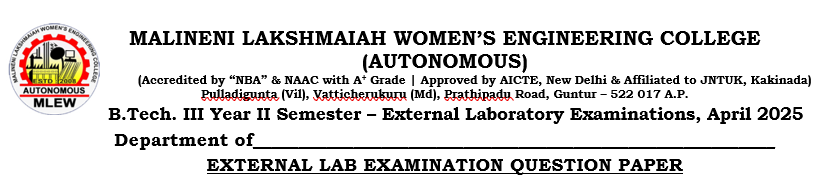
Set-1 **(CO1,CO4)(L3)(PO1, PO2, PO5)**

1. Create a website for Online shopping store say IEKart. Include the Metadata element in Homepage.html for providing description as "IEKart's is an online shopping website that sells goods in retail. This company deals with various categories like Electronics, Clothing, Accessories etc.”
2. Create a Typescript generic array and function to sort numbers as well as string values.



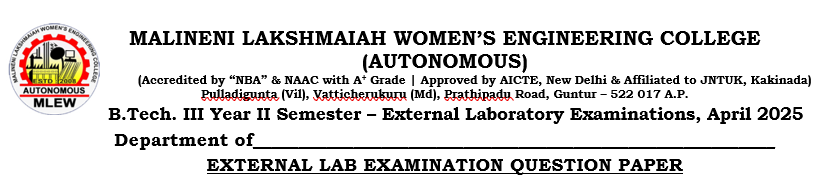
Set-2 **(CO1)(L3)(PO1, PO2, PO5)**

Using the form elements create Signup page for IEKart's Shopping application in html.



Set-3 **(CO2) (L3) (PO1, PO2, PO5)**

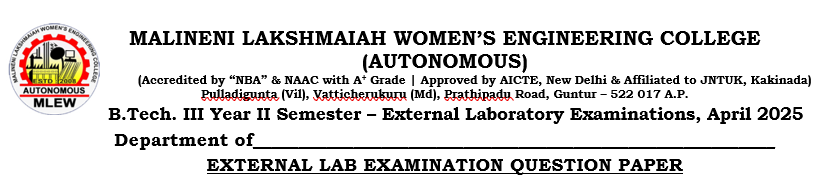
Create a file login.js with a User class. Create a validate method with username and password as arguments. If the username and password are equal it will return "Login Successful" else will return "Unauthorized access". Create an index.html file with textboxes username and password and a submit button. Add a script tag in HTML to include index.js file. Create an index.js file which imports login module and invokes validate method of User class. On submit of the button in HTML the validate method of the User class should be invoked. Implement the validate method to send the username and password details entered by the user and capture the return value to display in the alert.



Set-4 **(CO2, CO4) (L3) (PO1, PO2, PO5)**

a) Create an Employee class extending from a base class Person in javascript

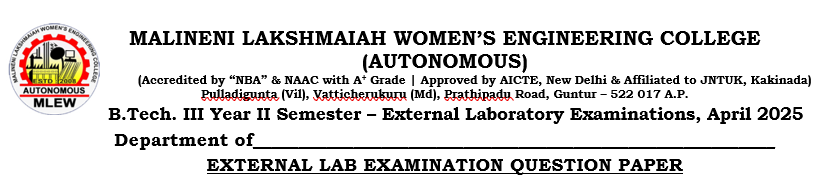
b) Design a TypeScript module for a Mobile Cart application that includes a function to determine the total cost of a product using the quantity and price values and assign it to a totalPrice variable.



Set-5 **(CO3, CO4) (L3) (PO1, PO2, PO5)**

a) Verify how to execute different functions successfully in the Node.js platform.

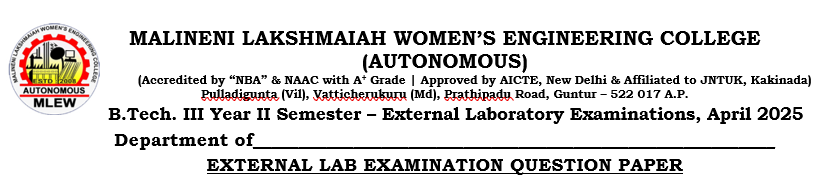
b) Create a namespace called ProductUtility and place the Product class definition in it. Import the Product class inside productlist file and use it.



Set-6 **(CO3, CO4) (L3) (PO1, PO2, PO5)**

a) Create a web server in Node.js

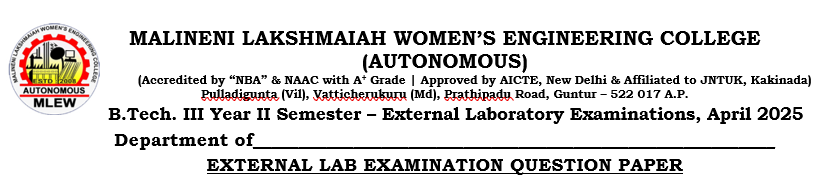
b) Create a Product class with properties namely productId and methods to setProductId() and getProductId()



Set-7 **( CO3, CO4) (L3) (PO1, PO2, PO5)**

a) Write a program to show the workflow of restarting a Node application.

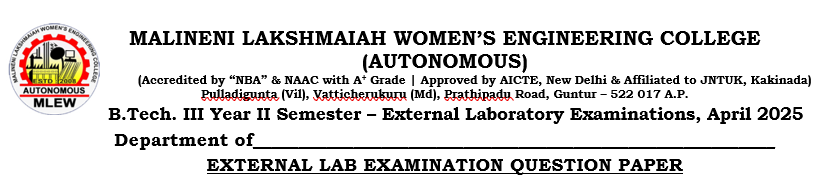
b) create a Product class with 4 properties namely productId, productName, productPrice, productCategory with private, public, static, and protected access modifiers and accessing them through Gadget class and its methods.



Set-8  **(CO3,CO4) (L3) (PO1, PO2, PO5)**

a) In node.js, Create a text file src.txt and add the following data to it. Mongo, Express, Angular, Node.

b) Declare a class named - Product with the below-mentioned declarations: (i) productId as number property (ii) Constructor to initialize this value (iii) getProductId method to return the message "Product id is <>".

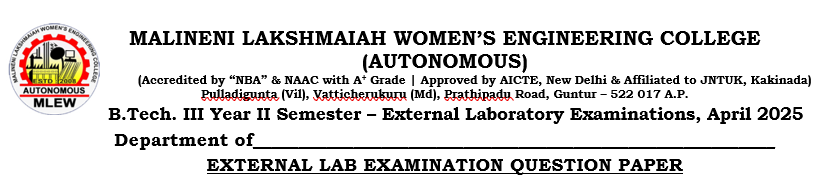


Set-9 **(CO4) (L3) (PO1, PO2, PO5)**

a) Consider that developer needs to declare a function - getMobileByManufacturer which

accepts a string as an input parameter and returns the list of mobiles.

b) Create objects of the Product class and place them into the productlist array



Set-10 **(CO4) (L3) (PO1, PO2, PO5)**

a) Consider that developer needs to declare a function - getMobileByManufacturer with two parameters namely manufacturer and id, where manufacturer value should pass as Samsung and the id parameter should be optional while invoking the function. If the id is passed as 101 then this function should return Moto mobile list. If manufacturer parameter is either Samsung/Apple then this function should return the respective mobile list, similar to making Samsung as default Manufacturer

b) Declare an interface named - Product with two properties like productId and productName with the number and string datatype and need to implement logic to populate the Product details. Use Duck Typing

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q.No. | Roll No. | | | | | | | | | | | | |
| Set1 | 501 | 511 | 521 | 531 | 541 | 551 | 561 | 571 | 581 | 591 | 5A1 | 5B1 |  |
| Set2 | 502 | 512 | 522 | 532 | 542 | 552 | 562 | 572 | 582 | 592 | 5A2 | 5B2 |  |
| Set3 | 503 | 513 | 523 | 533 | 543 | 553 | 563 | 573 | 583 | 593 | 5A3 | 5B3 |  |
| Set4 | 504 | 514 | 524 | 534 | 544 | 554 | 564 | 574 | 584 | 594 | 5A4 | 5B4 |  |
| Set5 | 505 | 515 | 525 | 535 | 545 | 555 | 565 | 575 | 585 | 595 | 5A5 | 5B5 |  |
| Set6 | 506 | 516 | 526 | 536 | 546 | 556 | 566 | 576 | 586 | 596 | 5A6 | 5B6 |  |
| Set7 | 507 | 517 | 527 | 537 | 547 | 557 | 567 | 577 | 587 | 597 | 5A7 | 5B7 |  |
| Set8 | 508 | 518 | 528 | 538 | 548 | 558 | 568 | 578 | 588 | 598 | 5A8 | 5B8 |  |
| Set9 | 509 | 519 | 529 | 539 | 549 | 559 | 569 | 579 | 589 | 599 | 5A9 | 5B9 |  |
| Set10 | 510 | 520 | 530 | 540 | 550 | 560 | 570 | 580 | 590 | 5A0 | 5B0 | 5C0 |  |