TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Java Assignments

Module – 2 (Core Java)

➔ Basic :

- Write a Java program to Take three numbers from the user and print the greatest number.
- Write a Java program that takes the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.
- Write a Java program that takes a year from user and print whether that year is a leap year or not. B19. Write a program in Java to display the first 10 natural numbers using while loop.
- Write a program in Java to input 5 numbers from keyboard and find their sum and average using for loop.
- Write a program in Java to display the pattern like right angle triangle with a number.

1

12

123

1234

12345

- Write a program in Java to make such a pattern like right angle triangle with number increased by 1 The pattern like:

1

2 3

4 5 6

7 8 9 10

Write a Java program that reads a positive integer and count the number of digits the number.

Input an integer number less than ten billion: 125463

Number of digits in the number: 6

- Write a Java program to print numbers between 1 to 100 which are divisible by 3, 5 and by both.

Array:

- Write a java program to find out the max number from given array using function
- Write a java program to find out the 2nd max number from given array using function.
- Write A Java Program to take two Array input from user and sort them in ascending or descending order as per user's choice
- Write a java program to make addition, Subtraction and multiplication of two matrix using 2-D Array
- Write a java program Find out length of string without using inbuilt function
- Write a java program to reverse a string and check that the string is palindrome or not.

OOPS :

- Create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of a rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square.
- Create a class with a method that prints "This is a parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call 1 - method of parent class by object of parent class 2 - method of child class by object of child class 3 - method of parent class by object of child class
- Create a class named 'Member' having the following members:
  1. Data members
  2. Name
  3. Age
  4. Phone number
  5. Address
  6. Salary

It also has a method named 'printSalary' which prints the salary of the members.

- Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize the length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super (s, s)'. Print the area and perimeter of a rectangle and a square.
- Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.
- Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.
- Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.
- Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC'are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.
- We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.
- Write a program to print the factorial of a number by defining a method named 'Factorial'. Factorial of any number n is represented by n! And is equal to 1*2*3*. *(n-1) *n.E.g.-

4! = 1*2*3*4 = 24

3! = 3*2*1 = 6 2! = 2*1 = 2 Also, 1! = 1 0! = 0

- We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

Write a program which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below:

Marks Grade :

1-100	A	
1-90	B	
1-80	B	
1-70	C	
1-60	D	
41-50	DD40	Fail

- Create a class named 'Shape' with a method to print "This is this is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

String:

- Write a Java program to count the letters, spaces, numbers and other characters of an input string.
- Write a Java program to print the ASCII value of a given character.
- Write a Java program that accepts an integer (n) and computes the value of n+nn+nnn. Input number: 5

5 + 55 + 555

- Write a Java program to display the system time.
- W.A.J.P to get the character at the given index within the String. Original String = Tops Technologies! The character at position 0 is T, The character at position 10 is o
- W.A.J.P to concatenate a given string to the end of another string.
- W.A.J.P to compare a given string to the specified character sequence. Comparing topsint.com and topsint.com: true Comparing Topsint.com and topsint.com: false
- W.A.J.P to check whether a given string ends with the contents of another string. "Java Exercises" ends with "se"? False "Java Exercise" ends with "se"? True
- W.A.J.P to check whether a given string starts with the contents of another string. Red is favorite color. Starts with Red? True Orange is also my favorite color. Starts with Red? False I3.
- W.A.J.P to find all interleaving of given strings.

The given strings are: WX YZ

The interleaving strings are: YWZX WYZX YWXZ WXYZ YZWX WYXZ

- W.A.J.P to find the second most frequent character in a given string. The given string is: successes The second most frequent char in the string is: c
- Create a class named 'Print Number' to print various numbers of different data types by creating different methods with the same name 'printn' having a parameter for each data type.
- Create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For

example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).

Exception :

- W.A.J. P to demonstrate try catch block,
- Take two numbers from the user and perform the division operation and handle Arithmetic Exception. O/P- Enter two numbers: 10 0

Exception in thread main java.lang.ArithmeticException:/ by zero

- W.A.J. P to demonstrate multiple catch blocks, (one is to handle divide by zero exception and another one is to handle

ArrayIndexOutOfBoundException) int a [] =new int [5]; a [5]=30/0;

- W.A.J. P to implement the above program (pro.no-B27) using nesting of try-catch block. try {

try

{//code}

catch (Exception e)

{//code}

catch (Exception e)

{//code}

- W.A.J. P to demonstrate try catch block, take two numbers from the user by Command line argument and perform the division operation and handle Arithmetic

O/P-

Exception in thread main java. Lang. Arithmetic Exception:/ by zero

- W.A.J.P to create the validate method that takes integer value as a parameter. If the age is less than 18, then throw an Arithmetic Exception otherwise print a message welcome to vote.

O/P- Enter your age: 16

Exception in thread main java. Lang. Arithmetic Exception: not valid

- W.A.J.P to create a custom exception if Customer withdraw amount which is greater than account balance then program will show custom exception otherwise amount

will deduct from account balance. Account balance is: 2000 Enter withdraw amount: 2500

Sorry, insufficient balance, you need more 500 Rs. To perform this transaction.

- W.A.J.P to create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

MultiThreading :

- W.A.J. P to create one thread by implementing Runnable interface in

Class.

- W.A.J. P to create one thread by extending Thread class in another Class.
- W.A.J.P to create 2 threads and execute that threads by providing sleep time as 2000ms and check the execution.
- W.A.J.P to start the same Thread twice by calling start () method twice. Test ThreadTwice1 t1=new TestThreadTwice1(); t1.start (); t1.start ();
- W.A.J.P to create 2 threads and make one thread as Daemon Thread by using set Daemon () method of Thread class and check whether the thread is set daemon or not by using is Daemon () method.

  TestDaemonThread2 t1=new TestDaemonThread2();

  TestDaemonThread2 t2=new TestDaemonThread2(); t1.start();

  t1.setDaemon(true);//will throw exception here t2.start();

Collection :

Write a Java program to create a new array list, add some colors (string) and print out the collection.

- Write a Java program to iterate through all elements in an array list.
- Write a Java program to insert an element into the array list at the first position.
- Write a Java program to retrieve an element (at a spec. index) from a given array list.
- Write a Java program to update specific array element by given element.
- Write a Java program to remove the third element from an array list.
- Write a Java program to search an element in an array list.
- Write a Java program to sort a given array list.
- Write a Java program to copy one array list into another.
- Write a Java program to shuffle elements in an array list.
- Write a Java program to append the specified element to the end of a hash set.

- Write a Java program to iterate through all elements in a hash list.
- Write a Java program to get the number of elements in a hash set.
- Write a Java program to associate the specified value with the specified key in a Hash Map.
- Write a Java program to count the number of key-value (size) mappings in a map.
- Write a Java program to reverse elements in an array list.
- Write a Java program to extract a portion of an array list.
- Write a Java program to compare two array lists.
- Write a Java program of swap two elements in an array list.
- Write a Java program to join two array lists.
- Write a Java program to convert a hash set to an array.
- Write a Java program to convert a hash set to a List/Array List.
- Write a Java program to check whether a map contains key-value mappings (empty) or not.
- Write a Java program to increase the size of an array list.
- Write a Java program to replace the second element of an Array List with the specified element.
- Write a Java program to print all the elements of an Array List using the position of the elements.
- Write a Java program to compare two sets and retain elements which are same on both sets.
- Write a Java program to get a collection view of the values contained in this map.

Module –3 (RDBMS & Database Programming With JDBC)

Field	Type	Null	Key	Default
Empno	int(4)	NO	PRI	0
Ename	varchar(10)	YES		(NULL)
Job	varchar(9)	YES		(NULL)
Mgr	int(4)	YES		(NULL)
Hiredate	date	YES		(NULL)
Sal	decimal(7,2)	YES		(NULL)
Comm	decimal(7,2)	YES		(NULL)
Deptno	int(2)	YES	MUL	(NULL)

DEPT TABLE:

Field	Type	Null	Key	Default
Deptno	int(2)	NO	PRI	0
Dname	varchar(14)	YES		(NULL)
Loc	varchar(13)	YES		(NULL)

STUDENT TABLE:

Field	Type	Null	Key	Default
Rno	int(2)	NO	PRI	0
Sname	varchar(14)	YES		(NULL)
City	varchar(20)	YES		(NULL)
State	Varchar(20)	YES		(NULL)

EMP_LOG TABLE:

Field	Type	Null	Key	Default
Emp_id	int(5)	NO		(NULL)

Log_date	Date	YES		(NULL)
New_salary	Int(10)	YES		(NULL)
Action	Varchar(20)	YES		(NULL)

DEPT TABLE DATA:

Deptno	dname	loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

EMP TABLE DATA:

Empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	(NULL)	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	(NULL)	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	(NULL)	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	(NULL)	10
7788	SCOTT	ANALYST	7566	1987-06-11	3000.00	(NULL)	20
7839	KING	PRESIDENT	(NULL)	1981-11-17	5000.00	(NULL)	10
7844	TURNER	SALESMAN	7698	1981-08-09	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1987-07-13	1100.00	(NULL)	20

7900	JAMES	CLERK	7698	1981-03-12	950.00	(NULL)	30
7902	FORD	ANALYST	7566	1981-03-12	3000.00	(NULL)	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	(NULL)	10

i. Select unique job from EMP table.
ii. List the details of the emps in asc order of the Dptnos and desc of Jobs?
iii. Display all the unique job groups in the descending order?
iv. List the emps who joined before 1981.
v. List the Empno, Ename, Sal, Daily sal of all emps in the asc order of Annsal.
vi. List the Empno, Ename, Sal, Exp of all emps working for Mgr 7369.
vii. Display all the details of the emps who's Comm. Is more than their Sal?
viii. List the emps who are either 'CLERK' or 'ANALYST' in the Desc order.
ix. List the emps Who Annual sal ranging from 22000 and 45000.
x. List the Enames those are starting with 'S' and with five characters.
xi. List the emps whose Empno not starting with digit78.
xii. List all the Clerks of Deptno 20.
xiii. List the Emps who are senior to their own MGRS.
xiv. List the Emps of Deptno 20 who's Jobs are same as Deptno10.
xv. List the Emps who's Sal is same as FORD or SMITH in desc order of Sal.
xvi. List the emps whose jobs same as SMITH or ALLEN.
xvii. Any jobs of deptno 10 those that are not found in deptno 20.
xviii. Find the highest sal of EMP table.
xix. Find details of highest paid employee.
xx. Find the total sal given to the MGR.
xxi. List the emps whose names contains 'A'.
xxii. Find all the emps who earn the minimum Salary for each job wise in ascending order.
xxiii. List the emps whose sal greater than Blake's sal.
xxiv. Create view v1 to select ename, job, dname, loc whose deptno are same.
xxv. Create a procedure with dno as input parameter to fetch ename and dname.
xxvi. Add column Pin with bigint data type in table student.

xxvii. Modify the student table to change the sname length from 14 to 40. Create trigger to insert data in emp_log table whenever any update of sal in EMP table. You can set action as 'New Salary'.

JDBC with Swing :

1. Write swing example with database connectivity to achieve the following.

Module – 4 (Web Technologies in Java)

- Write a Java program to fetch data from web.xml to Servlet using ServletConfig.
- Write a Java program to fetch data from web.xml to Servlet using ServletCotext.
- Write a Java program to submit student information (fname, lname, email, mobile, gender, password) using jsp form to servlet. Fetch data at servlet and print all the data in console.
- Write above Java program and print fetched data on another jsp using expression language.
- Write a Java program to fetch all the data from database table and print on jsp page using JSTL SQL tag library.
- Write a Java program to validate jsp form server side.

First Name: Only Alphabets Last Name: Only Alphabets Mobile: Only 10 Numbers Email: Standard Email Id

Password: Minimum One 1 Upper, Minimum 1 Lower, Minimum 1 Digit, Minimum 1 Special Character from @, #, $, %, _, & I2. Write CRUD operation using jsp only.

- Write a jsp/servlet CRUD operation for following.

Student:

- Int id;(primary key, Auto Increment)
- String fname, lname, email, mobile, gender, password;
- Need to use bootstrap responsive template for the same.
- Use client side validation to for all data input.
- Use regular expression for email for standard email input.
- Use regular expression for password like(Test@123)
- Use server side validation (Filter) same as client side validation.
- All the inserted data should be show in show.jsp with edit and delete functionality.
- Store all the deleted record in table named deleted data.

- Write a Java dynamic application “Message Passing System “using MVC and JDBC.

User:

Int uid ;(primary key, auto increment)

String fname, lname, email, mobile; Message:

int uid; String from, to, msg;

- Create one registration form to register user.
  i. Do client and server side validation. Use AJAX to register with unique email id.
  ii. After successful registration confirmation email should be sent to user's email id with one OTP.
  iii. Verify OTP and then and then allowed to login to the user. o After successful login user can edit their profile.
  iv. Also one user can send some msg to another user using their email id.
  v. When another user logged in they are able to see msg sent by a particular user and also can reply.
  vi. Logout button is there to invalidate session.
  vii. Also take care when user logged out and press back button on browser then it should be in logged out mode.
- W.A.J.P using above configuration with annotation and also show login jsp after data insert or registration.
- Write above Java program for after successful registration student can login and if login credentials are correct then show student's home page with his/her detail.

Module – 5 (Software Design and pattern)

- Write a program to create and read the cookie with any user define name enter into text box and go to servlet1 and again any support to button to go another servlet

Module – 6 (Hibernate Framework)

- Write a program to establish One to One Relationship between given 2 classes and perform a CRUD operation.

Student:
Int studentId; String studentName; Address; Address: Int addressId;
String street, city, state, zip code;

- Write a program to establish one to many and Many to One relationship between given 2 classes and perform a CRUD operation.

Cart: int cartId; double total;
String name; Set<Item> items;
Items:Int id;String itemId;Double
itemTotal; Int quantity; Cart cart;

- Write a program to establish many to many relationships between given 2 classes and perform a CRUD operation.

Reader:

Int readerId;

String email, firstName, lastName;

Set<Subscription> subscriptions; Subscription: Int subscriptionId;

String subscriptionName; Set<Reader> readers;

Module – 7 (Spring Framework)

- Write a program to demonstrate the setter based dependency injection.
- Write a program to demonstrate the constructor based dependency injection.
- Write a program to demonstrate the object based dependency injection and also implement inner bean concept in your spring beans configuration file.
- Write a program to collect 5 student information using spring collection(List) in spring bean configuration file.

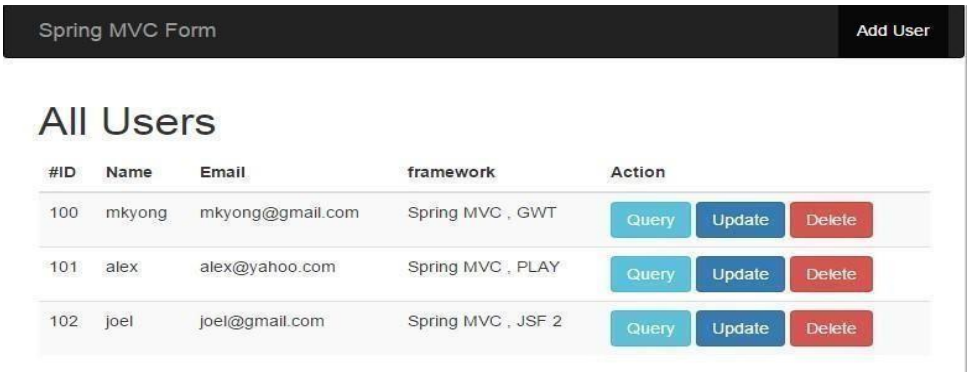
Student:

Int id; String fname, lname, email, mobile.

- Write a Spring ORM application to demonstrate following things.

2. Need to use bootstrap template.
3. There will be 2 modules. User and Admin.
4. User (id, firstname, lastname, email, mobile, password, gender, profile_pic) have to register to the site. (registration.jsp)
5. Then user have to login(login.jsp) and can able to edit profile like email, mobile or profile_pic.
6. Admin can login to the site and able see all the user registered with system and also able to delete profile of a user.

- Write a Spring MVC+ORM application to demonstrate the following



1. Add user with validation.
2. On clicking Query display single user data.
3. Update is for changes in user data.
4. Delete data.
5. Use bootstrap for UI part.

Module – 8 (Spring Boot)

1. What is Spring Boot and what are its Benefits?
2. Why Spring Boot is preferred over any other framework?
3. What are the key dependencies of Spring Boot?
4. What are the advantages of Spring Boot?
5. What are the features of Spring Boot?
6. How do you create a Spring Boot application using Maven?
7. what are the Spring Boot Annotations?
8. What are the Spring Boot properties?
9. What does REST stand for?
10. What is a resource?
11. What is the difference between @Controller and @RestController?
12. create springboot web app for product management :

    1 : create form to add product category

    2 : create from to add product with available category

    3 : view all product with category

    4 : view all product with category by pagination

Module – 9 (Spring Boot with WebServices)

1. What is WebServices and what are its Benefits?
2. How Many type of Webservices and write its details.
3. Explain key terminology of webservices
4. Explain architecture of webservices
5. What is restful webservics?
6. Difference between SOAP vs Restful.

7 : create restful service for given requirement :

1 : POST : add userdata with validation

2 : GET : view all userdata

3 : PUT : update userdata

4 : DELETE : data userdata

Module - 10 (Microservices)

1 How to configure microservices in Spring Boot?

2 Is Spring Boot suitable for microservices?

3 How do microservices work in Spring Boot?

4 How to communicate two microservices?

5 What is a load balancer in microservices?

6 What is Spring Boot microservices architecture?

7 Can 2 microservices have same database?

8 What is Eureka in microservices?

9 Is REST API a microservice?

10 Is every API a microservice?

11.Create a simple microservices-based system using Spring Boot that returns "Hello World" when accessed.

Requirements:

1. Create two separate Spring Boot projects:

   - hello-service (returns "Hello")

   - world-service (returns "World")

2. Use RestTemplate to communicate between services

3. Create a third project, api-gateway, that routes requests to the appropriate service

4. Use Eureka for service discovery

5. Run the applications and test the endpoint (/hello) using a tool like curl or Postman

12 Create a simple microservices-based system using Spring Boot that performs CRUD operations on a User entity.

Requirements:

1. Create two separate Spring Boot projects:

   - user-service (handles CRUD operations for Users)

   - user-client (consumes the user-service and exposes a RESTful API)

2. Use RestTemplate to communicate between services

3. Implement CRUD operations in the user-service using Spring Data JPA

4. Expose the CRUD operations as RESTful endpoints in the user-client

5. Run the applications and test the endpoints using a tool like curl or Postman

13. Build a simple CRUD API:

- Create a Spring Boot microservice that exposes a RESTful API for CRUD operations on a single entity (e.g., User, Product, etc.).

- Implement service layers, repositories, and entity models.

2. Design a microservice architecture:

- Divide a larger application into smaller, independent microservices (e.g., authentication, inventory management, payment processing).

- Define API contracts and communication patterns between services.

3. Implement service discovery and registration:

- Use Netflix Eureka to register and discover microservices.