

1. Flowchart

Assignment CW

1. Display sum of two integers.
2. Display numbers from 1 to 100.
3. Accept radius of a circle. Display area of a circle
4. Swap values of two integer variables using third variable.
5. Find out given number is palindrome or not.

Assignment HW

1. Find maximum number from given 3 numbers.
2. Find first 10 even numbers.
3. Create table of a number.
4. Find if given number is odd or even.
5. Find if given number is prime or not.
6. Calculate factorial of a number.
7. Count number of digits in any number.
8. Display all even numbers from 1 to 100.
9. Display all odd numbers from 1 to 100.
10. Accept starting number 'fromNum' and end number 'toNum' and find odd numbers starting from 'fromNum' till 'toNum'.
11. Accept length and breadth of a rectangle and find perimeter.

Assignments(Additional)

1. Generate Fibonacci series. 0,1, 1, 2, 3, 5,8 k. where $k < n$. n is entered by user
2. Generate Fibonacci series for first n terms.
3. Find sum of all digits of a number.

2. Java basics

Assignment CW

1. Write a Java program to enter two numbers and perform all arithmetic operations.
2. Write a Java program to enter length in centimeter and convert it into meter and kilometer.
3. Write a Java program to enter temperature in Fahrenheit(°F) and convert it into Celsius(°C)
4. Write a Java program to enter P, T, R and calculate Simple Interest.

Assignment HW

1. Write a Java program to enter two numbers and find their sum.
2. Write a Java program to enter length and breadth of a rectangle and find its perimeter.
3. Write a Java program to enter length and breadth of a rectangle and find its area.
4. Write a Java program to enter radius of a circle and find its diameter, circumference and area.
5. Write a Java program to enter temperature in °Celsius and convert it into °Fahrenheit.
6. Write a Java program to convert days into years, weeks and days.
7. Write a Java program to find power of any number x^y .
8. Write a Java program to enter two angles of a triangle and find the third angle.
9. Write a Java program to enter base and height of a triangle and find its area.

Assignments(Additional)

1. Write a Java program to calculate area of an equilateral triangle.
2. Write a Java program to enter marks of five subjects and calculate total, average and percentage.
3. Write a Java program to enter P, T, R and calculate Compound Interest.

3. Conditional statements

Assignment CW

1. Write a Java program to check whether a year is leap year or not
2. Write a Java program to check whether a number is divisible by 5 and 11 or not
3. Write a Java program to find maximum between three numbers
4. Write a Java program to input any alphabet and check whether it is vowel or consonant
5. Write a Java program to create Simple Calculator using switch case

Assignment HW

1. Write a Java program to check whether a number is negative, positive or zero
2. Write a Java program to input any character and check whether it is alphabet, digit or special character
3. Write a Java program to check whether a number is even or odd using switchcase

Assignments(Additional)

1. Write a Java program print total number of days in a month using switch case
2. Write a Java program to input basic salary of an employee and calculate its Gross salary according to following:
Basic Salary \leq 10000 : HRA = 20%, DA = 80%
Basic Salary \leq 20000 : HRA = 25%, DA = 90%
Basic Salary $>$ 20000 : HRA = 30%, DA = 95%
3. Write a Java program to input electricity consumption unit and calculate totalelectricity bill according to the given condition:
For first 50 units Rs. 0.50/unit
For next 100 units Rs. 0.75/unit
For next 100 units Rs. 1.20/unit
For unit above 250 Rs. 1.50/unit
An additional surcharge of 20% is added to the bill

4. Loops

Assignment CW

1. Display numbers from 1 to 100.
2. Display all even numbers from 1 to 100.
3. WAP to print following Pattern

```
* * * * *
* * * *
* * *
* *
*
```

4. WAP to print following Pattern

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

Assignment HW

1. Display the seventh element between 400 to 500 (both numbers excluded).
2. Find first 10 even numbers.
3. Find all prime numbers < 100.
4. Find all prime numbers from 400 till 300.
5. WAP to print following Pattern

```
*
* *
* * *
* * * *
* * * * *
```

6. WAP to print following Pattern

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
4 5
3 4 5
2 3 4 5
1 2 3 4 5
```

Assignments(Additional)

1. Calculate factorial of a number.
2. Count number of digits in any number.
3. Generate Fibonacci series. 0,1, 1, 2, 3, 5,8 k. where k < n . n is entered by user
4. Generate Fibonacci series for first n terms.

5. Find sum of all digits of a number.
6. Find out if given number is palindrome or not.
7. WAP to print following Pattern

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

8. WAP to print following Pattern

```
*
*
* *
* * *
* * * *
* * * *
* * *
* *
*
*
```

9. WAP to print following Pattern

```
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
```

10. WAP to print following Pattern

```
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
```

11. WAP to print following Pattern

```
1 1 1 1 1
1 1 1 2 2
1 1 3 3 3
1 4 4 4 4
5 5 5 5 5
```

12. WAP to print following Pattern

```
0 0 0 0 0
0 1 0 0 0
0 0 2 0 0
0 0 0 3 0
0 0 0 0 4
```

5. OOP Concepts

Assignments CW:

1. In Class
2. create class Student
3. create 2 objects of Student using new keyword
4. print 2 objects in sysout and see its hashCode is different Eg Student@15db9742 and Student@2329742
5. If you do this Student s1 = new Student(); Student s2 = s1;
 - a. Now print sop(s1) sop(s2) see both reference variables are pointing to same location.
 - b. But if you use new keyword for s1 and s2 see both values would be different
6. addid,name in student class. assign value to id name there itself and in main method print values. Then change value of instance variables and again print the values. Observe you cannot access id directly without creating object. Also 2 objects have 2 seperate set of data.
7. add method displayData inside Student class change values of id name in that method & also print the changed values in same method. Call displayData method from main method and see the output. Observe displayData can only be called by creating object of Student.

Assignments HW

1. Create getId() method in Student class. callgetId() method from displayData method so that you know that one method can be called from another method. Return id from getId() method and set that id to the instance variable in displayData method.
2. create another method in Student class setCustomData(intcustomId, String customName) These 2 variables are local variables. Call setCustomData from main method using Student object and assign any values to customId, customName from main method. Observe your values are overwritten in id and name instance variables. Then print id and name in main method.
3. Show java is pass by value and not by reference. To prove create object of Student. Pass that object to a method. Change value of id in that method but dont return anything. Also print student object hashCode see it will be same as that of main method object. Now after method is over in main method simply print id var and see value would be changed. This is because student reference points to a memory location. That location is passed as a value to the local variable of the calling method. Now since both reference variables are pointing to same location id is changed and hence the output.
4. From scanner In loop (for 2 iterations) read Student data create new object each time assign values and print student values in console.
5. Create Mydate class which have 3 instance variables (day, month, year)
6. Create two objects of Mydate and assign values to them, and display it in main method

8. Constructor

Assignments CW:

1. Create one class Dog and create only one default constructor. And create one object for Dog class in main() method.
2. Create one class Cat and create only one but parameterized constructor. And create one object like `Cat c = new Cat();` It will give error Since you have defined your own parameterized constructor, Java does not provide default one.
3. Create a class Cycle with member variables: `intaccountNo`, `intnoOfWheels`.
4. Create a default constructor with a SOP in it "I am default constructor"
5. Create another constructor which takes 2 arguments, calls the default constructor using `this()` and has a SOP in it "I am another constructor".
6. In main method, create an object of type Cycle by using default constructor. Note the output.
7. Create another object of type Cycle by using the parameterized constructor. Note the sequence of SOPs indicating that inner most ctor is called first.
8. Create class student and write a meaningful program with 2 constructors(default and parameterized), 3 methods (1 with return type (calculating percentage), 1 without return type (displaying student data), 1 with parameter list).

TQPP

6. Methods:

Assignments: (Divide CW/HW)

1. Write a program for class and object : (initialization through method)
Note: Initializing object simply means storing data into object
2. Write a program to demonstrate anonymous object.
3. Write a program to create multiple objects of one type only.
e.g. Rectangle r1=new Rectangle(), r2=new Rectangle();//creating two objects
4. Create Mydateclass which have 3 instance variables (day, month, year). Create two methods setDate() and displayDate() for Mydate class where write a logic for assigning some values to instance variable in setDate and display logic in displayDate() method.
5. Create Calculator class which have 4 methods add(inti, int j), sub(inti, int j), mul(inti, int j), div(inti, int j) with return type double.
6. Create one class Box having instance variables (height, width, depth). Add one method to calculate volume with void return type.
7. Create one class Box having instance variables (height, width, depth). Add one method to calculate volume with double return type.
8. Create Mydateclass which have 3 instance variables (day, month, year). Create two methods setDate() and displayDate() for Mydate class and accept Mydate values from user.
9. Create a class Account containing following methods :
insert() to insert account data
display() to display account information
deposit() to deposit amount
withdraw() to withdraw amount
checkbalance() to check balance
e.g.

```
class Account{
    intacc_no;
    String name;
    float amount;
}
```
10. Create Employee class likee.g.

```
classEmployee{
    intemp_id;
    String name;
    double salary;
}
```

Accept how many employees they want to create then accept that many values and display it.
Create appropriate methods for this
11. Write a program to copy values of one object into another by assigning the values of one object into another.
12. Write a program to copy values of one object into another by using method.

TQPP

7. Variables & OOP Concepts

Assignments CW:

1. Create one class Employee (emp_id, name, sal) with private access specifier and create getter and setter.
2. Create Academy class with appropriate attributes (instance variable) and create getter & setter
3. Write 3 Meaningful programs for single inheritance.
4. Write 3 Meaningful programs for multilevel inheritance.
5. Write 3 Meaningful programs for hierarchical inheritance
6. Create Class Laptop which has variables noOfUSBPort, processorSpeed of type int. Create getter, setter methods for the variables. In main method, 1> create Laptop object 2> set values of variables noOfUSBPort, processorSpeed using setter methods. 3> print variables noOfUSBPort, processorSpeed using getter methods.
7. Create class IPLTeam with method play. Create child classes of IPLTeam called as CSK, RCB. In main, call play from child class objects.
8. WAP to create a class Kid with method readBook() and another method readBook () with 2 parameters. The method readBook here is over-loaded (same method name but different parameters)
9. Create a class BigKid which extends Kid created above. Implement readBook() differently in BigKid class. Here the method readBook() has been over-ridden in the child class BigKid()
10. Create a class Teenager which extends Kid created above. Implement readBook() differently in Teenager class. In main method, declare 2 variables k1, k2 of type Kid. Create objects of type BigKid and Teenager such that K1 should reference object of class BigKid and K2 should reference object of class Teenager. Call their respective readBook() methods. The output is different from both the method calls even if the variable type is the same i.e. Kid. This is compile time polymorphism example.

11. Add method overloading

9. *this* & *super* keyword

Assignments CW:

1. Create a class Student with int id and String name as member variables.
2. Generate getter and setter methods for id and name. Observe in the generated setter method how the parameter's value is assigned to the member variable using this keyword.
3. What would happen in the setter method if you remove 'this.' when assigning the parameter value?
4. In main create 2 objects of Student.
5. Set id of first student to 11 and 21 to the other student's id
6. Check and observe if you can use 'this' in main method.
7. super can be used to refer immediate parent class instance variable.
8. super can be used to invoke immediate parent class method.
9. super() can be used to invoke immediate parent class constructor.
10. Create a class Cycle with member variables: intaccountNo, intnoOfWheels.
11. Create a default constructor with a SOP in it "I am default constructor"
12. Create another constructor which takes 2 arguments, calls the default constructor using this() and has a SOP in it "I am another constructor".
13. In main method, create an object of type Cycle by using default constructor. Note the output.
14. Create another object of type Cycle by using the parameterized constructor. Note the sequence of SOPs indicating that inner most constructor is called first.

10. static & final keywords

Assignments CW:

1. Create a class Car with String model and int totalCarSold as static variable in main() method.
2. Change the value of variable by class name then Print it.
3. Create two object of Car class and change value of static variable by 1st object and print it by accessing 2nd object.
4. Check following variations:
 - a. Static variable and Non-Static (Instance Variable) – Just declare and print it.
 - b. See if you can access static variable in static method
 - c. non static variable in static method
 - d. then static variable in non-static method
 - e. non static variable in non-static methodNote: non static means instance variable
5. See if you can overload static method, by using static keyword for 2nd overloaded method and without using static keyword.
6. See if you can override static method.
7. WAP to check whether you can inherit static variable of Parent class in Child class
8. WAP to check whether you can inherit static method of Parent class in Child class
9. WAP to Show 3 use of final - final class, final variable and final method
10. Create a class FinalDemo. In class Final Demo, declare a member variable x of type int. Declare it as final. In main method,
 - a. create an object of type FinalDemo. Try to change the value of x.
 - b. Now declare and implement a method method2 and make it final. Create a class FinalDemoChild which extends Final Demo. In FinalDemoChild, try to override the method method2
 - c. Now declare the class FinalDemoChild as final and try to extend the class FinalDemoChild
11. WAP to check whether final method of parent can be inherited in Child class.
12. WAP to check whether private method of parent can be inherited in Child class.

Assignments HW

1. Write a method to accept variable no of integers. Method should find sum of all the integers and display the result.
2. Write Employee Class with attributes like employee_id, name, and salary. Create parameterized constructor to initialize this 3 attributes, and write a method to display this information of 5 objects, display total no of employees using static method, totalEmp(), which will return total no of employees.
3. Write another same Employee class to implement auto-generation of employee_id.
4. Write a meaningful program to show the use of method overloading.
5. Write a meaningful program to show the use of method overriding.
6. WAP to show all the use of 'this' keyword.
7. WAP to show all the use of 'super' keyword.
8. WAP to show use of 'final' keyword (with variable, with method and with class).
9. WAP to use this and super in an instance method.

10. WAP to use this and super in static method.
11. WAP to show ways to access static variable
12. WAP which contain default constructor, parameterized constructor, one static method, one instance method, one static block and show the execution sequence.
13. WAP with 2 constructor (Default and parameterized), one instance method, one static method and check if final variable is not initialize at the time of declaration so where we can initialize them.
14. Write another program same as above but check final static variable.
15. WAP to show ambiguity in method overloading
16. WAP to overload static method.
17. WAP to override the static method.

TQPP

11. Abstract Class & Interface

Assignments CW:

1. Create an abstract class Machine with an implemented / concrete method rotate and an abstract method crush. Create a class Juicer which extends this abstract class Machine and implements method crush. Now, add another method filter in the class Juicer. In main method, Create an object of type Juicer and calls its crush, rotate and filter methods.
Create an object of type Juicer with reference variable of Machine (Machine m = new Juicer). Check the methods available to m.
2. Above we created an abstract class Machine with an abstract method crush and a concrete / implemented method rotate. Now create another abstract class Mixer which extends the abstract class Machine, has method crush implemented and has an additional concrete method blend. In main create an object of class of Mixer. Call methods rotate, blend and crush.
3. Create 2 abstract classes Abs1 and Abs2 each with different implemented methods doAbs1() and doAbs2() respectively. Check if you can create a class Temp which extends both the abstract classes. (Note- A class cannot extend 2 classes simultaneously)
4. Check following variations:
 - a. Can we create an object of an abstract class?
 - b. Can we declare a class abstract with no methods in it?
 - c. Can we declare a class abstract even if it does not have any abstract method?
 - d. Can we declare an abstract class which has both abstract as well as implemented methods?
 - e. Can a class extend 2 or more abstract classes?
 - f. Can an abstract class extend 2 or more abstract classes?
 - g. Can an interface extend 1 or many abstract classes?
 - h. Can an abstract class implements 1 or many interface?
5. WJJP2 create an abstract class Parent. Add an abstract method cook() in it which has only definition and one method wash() which has implementation. Create class Child which extends Parent and add the missing method implementation. In main, use both the methods by creating instance of the concrete class.
6. Show use of parameterized constructor of parent class (Child class invokes parent's parameterized constructor) using super
7. **Create interface Cake with a method declared as bake. Create 2 classes Strawberry, BlackForest both implementing Cake interface.??????????**
8. **Create interface IceCream with method eat and Juice with method drink. Create class Mastani which implements both interfaces.????????????????**

12. Containment (Object under object)

Assignments CW:

1. WAP to have Department class created with id, name . Student class has roll, name and Department object should have id and name. Assign and print individual values in main method
2. WAP to use containment for following hierarchy. Fill all data and print all data: (id is a integer number, name is a String)

Class Institute:

id name

Branch:

id name

Subject:

id name

Topic:

id name

SubTopic:

id name

Question:

id name

3. Create class Person which has attributes (name, gender, age, Address). Address is class which has attributes (city, state, country). Display Persons data
Note. Containment using constructor and getter/setter
4. Create Employee class which has attributes (id, name, salary, dept, mydate). Where dept and mydate is class, dept has (dept_id, dept_name) And mydate has (day, month, year).
Display Employee information.
Note. Containment using constructor and getter/setter

13. Access Modifiers

Assignments:

Create a class Student with 4 variables called as rollNo, admissionNo, age, courseId. Each of the variables should have one of the access modifiers: public, protected, no-access-modifier and private. Add 4 methods in the class: public method doPublic, <default> modifier method doDefault, protected method doProtected, private method doPrivate.

1. In main method outside the class but in same package – create object of type Student.
 - a. Try to access all the variables and all the methods in it. Verify the visibility against the access modifier table.
2. In same package as Student, create a new class TQPPStudent which extends the Student class.
 - a. Create a method called as checkStudentVariableAccess in TQPPStudent and try to assign values to the inherited variables in the method. Verify the visibility against the access modifier table.
 - b. Create a method called as checkStudentMethodAccess in TQPPStudent and try to call the inherited methods in the method. Verify the visibility against the access modifier table.
3. In another package, create a new class OtherPackageStudent which extends the Student class.
 - a. Create a method called as checkStudentVariableAccess in Student and try to assign values to the inherited variables in the method. Verify the visibility against the access modifier table.
 - b. Create a method called as checkStudentMethodAccess in Student and try to call the inherited methods in the method. Verify the visibility against the access modifier table.
4. In main method outside the class but in some other package – create object of type Student.
 - a. Try to access all the variables and all the methods in it. Verify the visibility against the access modifier table.
5. In main method outside the class but in some other package – create object of type TQPPStudent.
 - a. Try to access all the variables and all the methods in it. Verify the visibility against the access modifier table.
6. In main method outside the class but in some other package – create object of type OtherPackageStudent.
 - a. Try to access all the variables and all the methods in it. Verify the visibility against the access modifier table.

14. Arrays

Assignment CW

1. WAP to add elements to single dimensional int array and print elements from 1D array
2. WAP to accept dimension of 1D array and create and accept data in that array. Calculate the average value of array elements.
3. WAP to accept data in 2D array of 3X4 and accept data.
4. WAP to accept data in 2D array where rows are fixed and columns are variable.
5. WAP to search for a given number in an array and accordingly print the index if exists.
6. WAP to copy an array by iterating the array.
7. WAP to print reverse of an array. Also print every alternate element backwards.
8. Write two methods that return the average of an array with following headers.
 - a. `public static int average(int[] array)`
 - b. `public static double average(double[] array)`.
 - c. Write main and invoke the 2 methods.
9. WAP to insert an element in a specific position into an array.
10. WAP to print all negative elements in an array and also count total number of negative elements in an array.
11. WAP to put even and odd elements of array in two separate arrays.
12. WAP to find the maximum and minimum value in an array.
13. WAP to find and count total number of duplicate elements in an array.
14. WAP to print all unique elements in the array.
15. WAP to find the second smallest element in an array.
16. WAP to count frequency of each element in an array.
17. WAP to merge 2 arrays to 3rd array. 3rd array should not have elements of same value.
18. WAP sort array elements in ascending order.
19. WAP to print the employees from Employee[] array who has same salary (Create Employee class which has 3 attributes id, name, salary and add employee objects to your array)
20. WAP to reverse the array itself, don't print array in reverse – I want current array reverse. Means e.g. `arr[] = [3, 90, 45, 29, 37, 78]` so your same array must be `[78, 37, 29, 45, 90, 3]` without using temporary array.

Assignment HW

1. Create Stack implementation of array(push,pop,display)
2. Create Queue implementation of array.
3. WAP to reverse the array itself, don't print array in reverse – I want current array reverse. Means e.g. `arr[] = [3, 90, 45, 29, 37, 78]` so your same array must be `[78, 37, 29, 45, 90, 3]` by using temporary array.
4. Write a Java program to test the equality of two arrays. Means e.g. `arr1[] = [12, 22, 32, 42, 52, 62]` and `arr2[] = [52, 22, 62, 12, 42, 22]` Here both arrays are equal.
5. Write a Java program to find a missing number in an array. Means e.g. array has 1 to n element in sequence(n can be 50 or 100) `arr[] = [1, 2, 3, 4, 6, 7]` So missing no is 5.
6. WAP to replace all the 0's with 1's in your array. Your array is `[26, 0, 67, 45, 0, 78, 54, 34, 10, 0, 34]`.

7. WAP to replace all negative value with its immediate left elements square. Means `arr[] = [12, 3, -19, 29, 5, -61, 44, 7, -9]` Output array will be `[12, 3, 9, 29, 5, 25, 44, 7, 49]`.
8. WAP to arrange the elements of an given array of integers where all negative integers appear before all the positive integers.
9. WAP to arrange the elements of an given array of integers where all Even integers appear before all the Odd integers.
10. WAP to find min character in character array. Means e.g. `arr[] = ['A', 'D', 'E', 'x', 'z', 'R']`, so min character is 'A'.
11. WAP to display all perfect square number in array. Means e.g. `arr[] = [23, 43, 25, 49, 12, 9, 78, 66, 39, 0]` so output is 25, 49, 9.
12. WAP to print the employees from `Employee[]` array who has same joining date. You have `Employee` class which has 4 attributes `id`, `name`, `salary`, `date` (`date` is another object which has 3 attributes `day`, `month`, `year`) and add employee objects to your array
13. Create class `Dept(did, dname)`, class `MyDate(day, month, year)`
Class `Employee(emp_id, emp_name, salary, date(object), dept(object))`. Create array of `Employee` and display the array elements.
14. Same as above but print Employees whose `dept_name` is same.
15. WAP to print maximum in rowwise in 2D array. Means e.g. `arr[][] = {{22, 31, 9}, {12, 25, 16}}` output is: 31 and 25.
16. WAP to print minimum in columns. Means e.g. `arr[][]={{22, 31, 9}, {12, 5, 16}, {34, 42, 2}}` output is: 12, 5, 2.
17. WAP to shuffle array. Means e.g. `arr[] = [5, 6, 23, 67, 39, 10, 2]`
So output array is `[6, 23, 67, 39, 10, 2, 5]`.
30. WAP to show 3 dimension array.

Assignment (Additional)

1. WAP to left rotate an array.
2. WAP to right rotate an array.
3. WAP to print outer elements of 2D array of 4X4.
4. WAP to check if a matrix is a Sparse matrix (in which most of the elements are 0). Print "Sparse" if it is Sparse else print "Not sparse"
5. WAP to create transpose of a matrix (transpose is converting rows to columns) and print it.
6. WAP to subtract two matrices.
7. WAP to find sum of each row and column of a matrix.
8. WAP to find sum of main diagonal elements of a matrix.
9. WAP to find the average of the inner most elements of an array.
10. WAP to print upper triangular matrix.
11. WAP to find sum of lower triangular matrix.
12. Create an array of character to store an infix expression and convert to postfix expression
(`A+B/C`)=`ABC/+`

15. STRING

Assignment CW

1. String create with new operator and without new
2. Compare string using == when we create without new see same reference is there
3. Compare string using new operator when new is present see different reference is there.
4. What is String constant pool and Heap? Explain
5. Write a Java program to find length of a string.
6. Write a Java program to concatenate two strings.
7. Write a Java program to compare two strings.
8. Write a Java program to convert lowercase string to uppercase.
9. Write a Java program to copy one string to another string.
10. WAP to split string into 2 tokens where string is "HELLO\$WORLD"
11. Write a Java program to find first occurrence of a character in a given string.
12. Write a Java program to count occurrences of a character in given string.
13. Write a Java program to trim leading white space characters in a string.
14. Write a Java program to count total number of words in a string.
15. Write a Java program to find first occurrence of a word in a given string.
16. Write a Java program to search all occurrences of a character in given string.
17. Write a Java program to convert uppercase string to lowercase.
18. Write a Java program to remove all occurrences of a character from string.
19. Write a Java program to trim trailing white space characters in a string.
20. Write a Java program to trim both leading and trailing white space characters in a string.
21. Write a Java program to remove all extra blank spaces from a given string.
22. Write a Java program to toggle case of each character of a string.
23. Write a Java program to count total number of vowels and consonants in a string.
24. Write a Java program to find reverse of a string.
25. Write a Java program to reverse order of words in a given string.
26. Write a Java program to find last occurrence of a character in a given string.
27. Write a Java program to find highest frequency character in a string.
28. Write a Java program to find lowest frequency character in a string.
29. Write a Java program to count frequency of each character in a string.
30. Write a Java program to remove first occurrence of a character from string.
31. Write a Java program to remove last occurrence of a character from string.
32. Write a Java program to remove all repeated characters from a given string.
33. Write a Java program to replace first occurrence of a character with another in a string.
34. Write a Java program to replace last occurrence of a character with another in a string.
35. Write a Java program to replace all occurrences of a character with another in a string.
36. Write a Java program to find last occurrence of a word in a given string.
37. Write a Java program to search all occurrences of a word in given string.
38. Write a Java program to count occurrences of a word in a given string.
39. Write a Java program to remove first occurrence of a word from string.
40. Write a Java program to remove last occurrence of a word in given string.
41. Write a Java program to remove all occurrence of a word in given string.
42. Write a Java program to find total number of alphabets, digits or special character in a string.

43. Write a program using Regular Expression to check valid mobile number or not .Use matches method

Assignment (H.W)

1. Write a java program to find the duplicate words and their number of occurrences in a string
2. Write a java program to count the number of words in a string
3. Write a java program to check whether two strings are anagram or not?
4. Write a java program to reverse a given string with preserving the position of spaces
5. WAP to find longest word in the given sentence
6. How do you swap two string variables without using third or temp variable in java
7. Write a java program to find all permutations of a string
8. Accept email_id from user and check valid or not(should contain @,.)
9. Accept sentence replace each vowel by next consecutive character
10. Create an array of 10 names sort in descending order

17. Enum

Assignments:

1. Create enum seasons { SPRING , SUMMER, AUTUMN } .
In main method create enum variable for any of above enum constant and display its value.
2. Create instance variable in enum.
3. Create method in enum. Access variable and method in main method.
4. Create constructors in enum. Create all three objects using 3 different constructors in enum in main method.
5. Create two objects of type enum and compare them.
6. Use enums in switch statement
7. Create public constructors in enum. And try to create object of enum with new keyword in main method.
8. Implement an interface in enum.
9. Iterate over all values of enum.
10. Override method in enum.
11. Try to extend a class in enum.
12. Create duplicate constants in enums.

18. THREAD

Subtopics/Assignments:

1. Create thread by extending Thread class.
2. Create thread by implementing runnable interface.
3. Show example in which two threads share same object.
4. Create a daemon thread. Show in example that jvm shuts down even if daemon thread is running.
5. Change priorities of thread. Check what are values for MIN_PRIORITY, NORM_PRIORITY, MAX_PRIORITY.
6. What is default priority of a thread.
7. WAP to show use of sleep method
8. Write at least 3 differences between sleep and wait method.
9. Thread t1 prints numbers 1 to 10 . Thread t2 prints characters a to h. Ensure that always characters are printed first and then numbers using join method.
10. Show use of all 3 join methods of thread class.
11. Which methods are final , yield , sleep or join.
12. What is synchronization?
13. Can we use synchronized word with class and variable.
14. Show example of synchronized method in which two threads are trying to update same thread.
15. Show one example of synchronized block.
16. How to create class level lock in synchronized method. Show in your code.
17. How to create class level lock in synchronized block. . Show effect of synchronization if both threads are using different objects.
18. Show use of synchronization if two threads are sharing same objects.
19. If thread t1 is accessing static synchronized method m1 , can thread t2 access synchronized method m2 at same time.
20. What is advantage of using synchronized block over synchronized method.
21. Prove following statement in your code. While a thread executing static synchronized method, the remaining threads are not allowed to execute any static synchronized method of that class simultaneously. But remaining threads are allowed to execute the following methods simultaneously 1. Normal static methods 2. Synchronized instance methods 3. Normal instance methods.
22. A consumer threads consumes chocolates from a basket, producer thread produces fixed number of chocolates at a time. Write a program in which consumer thread checks for sufficient chocolates in basket, it waits for producer to produce if sufficient chocolates are not available in the basket and then consumes given number of chocolates. Producer thread will notify consumer thread after it finishes producing chocolates.
23. What is difference between notify and notify all.
24. Create a thread pool of 3 threads using executor framework. Create 10 threads which will run by reusing threads in thread pool.
25. What is deadlock. Show one example of deadlock.
26. Show use of interrupt method of thread class.

TQPP

19. INNER CLASSES

Assignments

1. Create class Outer in which you will have private , public , static and static final variables. Create static and non static inner class in 'Outer' class. Try to access all variables in both inner classes and check which are not accessible.
2. Create objects on static and non static inner classes in main method. Access methods ,variable of these classes in main method.
3. Create static and non static variables in both above inner classes.
4. Try to create static method in non-static inner class.
5. Show one example of method local inner class. Try to call method of inner class from main method.
6. Create anonymous class by implementing interface.
7. Create anonymous class by extending abstract class.
8. Create anonymous class by extending concrete class.
9. Can we implement multiple interfaces while creating anonymous class ?
10. Predict and study output of following program.

```
public class ShadowingExample {  
    public int x = 0;  
    static int num ;  
    class FirstLevel {  
        public int x = 1;  
        void methodInFirstLevel(int x) {  
            ShadowingExample st2 = new ShadowingExample();  
            System.out.println("x = " + x);  
            System.out.println("this.x = " + this.x);  
            System.out.println("ShadowingExample.x = " + st2.x);  
            System.out.println("ShadowingExample.num = " + ShadowingExample.num);  
        }  
    }  
    public static void main(String... args) {  
        ShadowingExample st = new ShadowingExample();  
        ShadowingExample.FirstLevel fl = st.newFirstLevel();  
        fl.methodInFirstLevel(23);  
    }  
}
```

11. Predict and study output of following program.

```
class OuterClass  
{  
    int x = 10;  
    void methodOfOuterClass()  
    {  
        System.out.println("From OuterClass");  
    }  
    class InnerClass  
    {  

```

```
int y = 20;
}
}
class AnotherClass extends OuterClass{
class AnotherInnerClass extends InnerClass{
}
}
public class InnerInheritance
{
public static void main(String args[]) {
AnotherClass anotherClass = new AnotherClass();
System.out.println(anotherClass.x);
anotherClass.methodOfOuterClass();
AnotherClass.AnotherInnerClass anotherInnerClass = anotherClass.new AnotherInnerClass();
System.out.println(anotherInnerClass.y);
}
}
```

12. Can you have more than one object of anonymous class?
13. Check the scope of inner class by changing access modifiers of inner class to public , private, protected.

20. DESIGN PATTERNS

Assignment:

1. What are different types of design patterns?
2. WAP to show a real world example of Factory Method Pattern?
3. WAP to show a real world example of Abstract Factory Pattern?
4. WAP to show a real world example of Singleton Pattern?
5. WAP to show Prototype Design Pattern?
6. WAP to explain Observer and Decorator Design Pattern?

TQPP

Data structures

TQPP

21.Collection(LinkedList)

Assignments CW :-

1. WAP to append the specified element to the end of a linked list.
2. WAP to iterate through all elements in a linked list.
3. WAP to iterate through all elements in a linked list starting at the specified position.
4. WAP to iterate a linked list in reverse order.
5. WAP to insert the specified element at the specified position in the linked list.
6. WAP to insert elements into the linked list at the first and last position.
7. WAP to insert the specified element at the front of a linked list.
8. WAP to insert the specified element at the end of a linked list.
9. WAP to insert some elements at the specified position into a linked list.
10. WAP to get the first and last occurrence of the specified elements in a linked list.
11. WAP to display the elements and their positions in a linked list.
12. WAP to remove a specified element from a linked list.
13. WAP to remove first and last element from a linked list.
14. WAP to remove all the elements from a linked list.

Assignments HW:-

1. WAP to shuffle the elements in a linked list.
2. WAP to join two linked lists.
3. WAP to clone a linked list to another linked list.
4. WAP to remove and return the first element of a linked list.
5. WAP to retrieve but does not remove, the first element of a linked list.
6. WAP to retrieve but does not remove, the last element of a linked list.
7. WAP to check if a particular element exists in a linked list.
8. WAP to convert a linked list to array list.
9. WAP to compare two linked lists.
10. WAP to test a linked list is empty or not.
11. WAP to replace an element in a linked list.

22. Collection (Arraylist)

Assignments CW:-

1. WAP to add elements to arraylist without using generics (no <>) and print content of it where Integer is used. In second arraylist String is used.
2. WAP add elements to arraylist without using generics, 0th location keep Integer, 1st location String now print each element and display contents.
3. WAP to use add operation of ArrayList
4. WAP to print all elements of ArrayList using iterator
5. WAP to iterate through all elements in an ArrayList using for loop
6. WAP to iterate through all elements in an ArrayList using for each
7. WAP to create a new ArrayList, add some colors (string) and print the collection.
8. WAP to remove element from ArrayList
9. WAP to remove all elements from ArrayList
10. WAP to retain all elements from ArrayList
11. WAP to know how many elements in ArrayList
12. WAP to test an ArrayList is empty or not
13. WAP to empty ArrayList
14. WAP to search an element from ArrayList
15. WAP to search the specified collection in this collection
16. WAP to retrieve an element (at a specified index) from a given ArrayList
17. WAP to insert an element into the ArrayList at the first position
18. WAP to add 1 to 50 numbers in ArrayList and print only even numbers (using iterator)
19. WAP to match two collections
20. WAP to check if collection is empty
21. WAP to convert collection into array
22. WAP of swap two elements in an ArrayList
23. WAP to replace the second element of an ArrayList with the specified element

Assignments HW:-

1. WAP to create Emp (id,name,sal) object and add 2objects to ArrayList. Sysout and see both variable memory space is printed. This is because toString is not overridden but if you would have done this for Integer then beautiful output would have been printed.
2. Now override toString for earlier assignment and now sysout and see values are printed
3. WAP to print Emp whose salary is > 10000
4. WAP to print Emp who have name "Sachin"
5. WAP to print Emp who have highest number of salary
6. WAP to use add all elements to ArrayList
7. WAP to sort a given ArrayList
8. WAP to copy one ArrayList into another
9. WAP to compare two ArrayLists print if equal?
10. WAP to join two ArrayLists
11. WAP to clone an ArrayList to another ArrayList
12. WAP to empty an ArrayList
13. WAP to trim the capacity of an ArrayList the current list size

14. WAP to increase the size of an ArrayList
15. WAP to update specific array element by given element
16. WAP to remove the third element from an ArrayList
17. WAP to shuffle elements in an ArrayList
18. WAP to reverse elements in an ArrayList
19. WAP to extract a portion of an ArrayList
20. WAP to print all the elements of an ArrayList using the position of the elements

Assignments(Additional):-

1. Show in program, that size of an arraylist dynamically changes
2. Create aarraylist of integers. Add first 50 odd numbers to arraylist.
3. Create aarraylist of string. Add 7 days to list (Monday , Sunday etc) Remove elements from list for which string length is more than 7
4. Create arraylist of 10 integers. Insert an element at 7th position.
5. Create arraylist of characters. Remove duplicate characters from the list. E.g. list – { 'd' , 'g' , 'h' , 'd' , 'd' . 'h' , 'g'}Output should be {'d' , 'g' , 'h'}
6. Let employee class have a department object. Iterate through aarraylist of employees.
7. Use copy method of collections class to make a copy of given arraylist of employees. Check if it creates shallow copy or deep copy. Check if department objects are also copied.
8. Sort arraylist of Strings.
9. Sort arraylist of employees on employee names using comparable interface.
10. Sort arraylist of employees on department names using comparator interface.
11. Create arraylist 'arrList' within an arraylist. Use contains and containsall method to check if given element and arraylists are present in 'arrList'
12. Iterate through arraylist using foreach.
13. Does ArrayList allow Null values
14. What is the difference between ArrayListclear() and removeAll() methods?
15. What is the difference between ArrayList's overloaded remove() methods?
16. If an ArrayList contains duplicate objects and remove() method is invoked on the same object, will itremove the duplicates?
17. How to reverse an ArrayList in Java?
18. How to prove ArrayList increases its internal space, prove by showing Java code?

23. Collection Sorting

Assignments CW:-

1. WAP to sort the elements of List that contains String objects. Print ArrayList. Sort using
2. Collections.sort(list) method. Print ArrayList.
3. WAP to sort the elements of List that contains Integer objects. Print ArrayList. Sort using
4. Collections.sort(list) method. Print ArrayList.
5. WAP to create a class Student with (rollNo, name and age). Create 3 Comparator implementations for each Student attribute (i.e. rollNo, name and age)
6. Create and add 4 Student objects in ArrayList. Print ArrayList.
 - a. Sort the list using rollno comparator. Print ArrayList.
 - b. Sort the list using name comparator. Print ArrayList.
 - c. Sort the list using age comparator. Print ArrayList.
 - d. Observe the sorted outputs in case the name, age and name + age are same.

Assignments HW:-

1. Sort arraylist of employees in ascending order of their salaries. If salary is same , list should be in descending order of name.
2. What will happen if compare method returns only +1. Show example.
3. What will happen if compare method always returns -1. Show example
4. What will happen if compare method always returns 0; Show example.
5. Reverse an arraylist of 10 integers.
6. Sort arraylist of integers without using sort method.

24. Collection Map

Assignments CW:-

1. WAP to add elements to a HashMap without using generics (ie do not use <>) and print content of it. Use Integer as Key and String as Value. In second HashMap add elements of String type as Key and Integer as Value.
2. WAP add elements to HashMap without using generics, 0th location use String as key and Integer as value, on 1st location use String as key String and Integer as value.
3. WAP to add elements to a HashMap using generics with Integer as Key and String as value. And 4 key-value Map.Entry
4. WAP to create a map using Wrapper class as key and value same as keys' data type WAP to count the number of key-value (size) mappings in a map.
5. WAP to get all the entries from a HashMap. Iterate the entries and print the Key & Value values
6. WAP to get only the Keys from a HashMap
7. WAP to get only the Values from a HashMap
8. WAP to create a LinkedHashMap which contains Integers as key and Strings as value. Print the map contents and observe the order. Don't use generics.
9. WAP to create a LinkedHashMap which contains Strings as key and Integers as value. Print the map contents and observe the order. Don't use generics.
10. WAP to create a LinkedHashMap which contains Strings as key and objects of different classes as value. Print the map contents and observe the order.
11. WAP to create a LinkedHashMap which contains Integers as key and Strings as value. Print the map contents and observe the order. Use generics.
12. WAP to create a LinkedHashMap which contains Strings as key and Integers as value. Print the map contents and observe the order. Use generics.
13. WAP to create a LinkedHashMap. Add 3 objects in it.
 - a. WAP to search a value in the Map.
 - b. WAP to get all keys in the Map.
 - c. WAP to remove an element from the Map.
14. WAP to create a TreeMap which contains Integers
15. WAP to create a TreeMap which contains Strings
16. WAP to search a key in a TreeMap
17. WAP to search a value in a TreeMap
18. WAP to get all keys from the given a TreeMap
19. WAP to delete all elements from a given TreeMap
20. WAP to copy a TreeMap content to another TreeMap
21. WAP to sort keys in TreeMap by using comparator
22. WAP to create a TreeMap with Integer as key and get a key-value mapping associated with the greatest key and the least key in a map
23. WAP to get the first (lowest) key and the last (highest) key currently in a TreeMap
24. WAP to get a reverse order view of the keys contained in a given map TreeMap
25. WAP to get a key-value mapping associated with the greatest key less than or equal to the given key
26. WAP to get the greatest key less than or equal to the given key

Assignments HW:-

1. WAP to copy all of the mappings from the specified HashMap to another map
2. WAP to test a HashMap is empty or not
3. WAP to test if a HashMap contains a mapping for the specified key
4. WAP to search for an element from HashMap using key
5. WAP to test if a HashMap contains a mapping for the specified value
6. WAP to remove an element from HashMap using key
7. WAP to empty Map. (Hint - Get keys and iterate through the keys to remove entries from Map)
8. WAP to create a map using Custom class as key and any other JDK provided object as value
9. WAP to create a map using Custom class as key and any other Custom class as value
10. WAP to create a map which stores an arraylist as a value
11. WAP to iterate the elements in the arraylist stored in the above Map.
12. WAP to create a Map which stores another Map as a value.
13. WAP to iterate the elements in the Map stored in the above Map
14. WAP to copy a HashMap content to another LinkedHashMap
15. WAP to delete all elements from a given LinkedHashMap
16. WAP to create a LinkedHashMap with Integer as key and any other object as value. Now get a key-value mapping associated with the highest key and the least key in a map
17. WAP to get the portion of a HashMap whose keys are strictly less than a given key
18. WAP to get the portion of this HashMap whose keys are less than (or equal to, if inclusive is true) a given key
19. WAP to get the least key strictly greater than the given key. Return null if there is no such key
20. WAP to get a key-value mapping associated with the greatest key strictly less than the given key. Return null if there is no such key
21. WAP to get the greatest key strictly less than the given key. Return null if there is no such key.

Assignments(Additional):-

1. Create class Department (contains id, name, listofStudents)
 - a. Create class Student (contains rollNo and Name)
 - b. Create 4 Departments objects with different ids and add n number (atleast 4) ofStudents to each of the dept.
 - c. Create a Map to add all the 4 Department objects with Dept id as key and Deptobj as value.
 - d. Now iterate through each of the dept, print its id and name and the Students within it (ie the rollNo and name).
2. (continuation of 1)
 - a. Create 1 Department object with already used id and add Students to the dept.
 - b. Add the Department object with Dept id as key and Deptobj as value.
 - c. Observe how many elements are present in the Map. Iterate through the elements in the Map.
3. (continuation of 2)
 - a. Over-ride hashCode method from Object class in the Department class
 - b. Create 1 Department object with already used id and add Students to the dept.
 - c. Add the Department object with Dept id as key and Deptobj as value.
 - d. Observe how many elements are present in the Map. Iterate through the elements in the Map.
4. (continuation of 3)

- a. Over-ride equals method from Object class in the Department class
 - b. Create 1 Department object with already used id and add Students to the dept.
 - c. Add the Department object with Dept id as key and Deptobj as value.
 - d. Observe how many elements are present in the Map. Iterate through the elements in the Map.
5. (continuation of 4)
 - a. Over-ride hashCode and equals methods from Object class in the Department class
 - b. Create 1 Department object with already used id and add Students to the dept.
 - c. Add the Department object with Dept id as key and Deptobj as value.
 - d. Observe how many elements are present in the Map. Iterate through the elements in the Map.
6. What do you conclude from exercise 1
7. WAP to get the portion of a HashMap whose keys range from a given key (inclusive), to another key (exclusive).
8. WAP to get the portion of a HashMap whose keys range from a given key to another key.
9. WAP to get a portion of a HashMap whose keys are greater than or equal to a given key.
10. WAP to get a portion of a HashMap whose keys are greater than to a given key.
11. WAP to get a key-value mapping associated with the least key greater than or equal to the given key. Return null if there is no such key.
12. WAP to get the least key greater than or equal to the given key. Returns null if there is no such key.
13. WAP to create a TreeMap to store the Employee objects in a descending order based on Employee Id
14. WAP to create a TreeMap to store the Employee objects in a descending order based on Employee Id and yet retrieve in ascending order
15. Create a class Employee which implements comparable in ascending order. WAP to create a TreeMap to store the Employee and stores objects in a TreeMap in descending order.

25. Collection Set

Assignments CW:-

1. WAP to create a HashSet with Integer objects without using generics
2. WAP to create a HashSet with some colors (String) using generics
3. WAP to create a HashSet from an ArrayList
4. WAP to iterate through all elements in a HashSet and print the elements. Observe the order of elements.
5. WAP to get the number of elements in a HashSet.
6. WAP to empty a HashSet.
7. WAP to test if a HashSet is empty or not.
8. Do all above assignments using LinkedHashSet
9. WAP to create a TreeSet from a HashSet.
10. WAP to create a new TreeSet, add Strings and print the TreeSet.
11. WAP to iterate through all elements in a TreeSet.
12. WAP to get the first and last elements in a TreeSet.
13. WAP to get the number of elements in a TreeSet.
14. WAP to create a reverse order view of the elements contained in a given TreeSet.
15. WAP to remove a given element from a TreeSet.
16. WAP to retrieve and remove the lowest element of a TreeSet using a single method call. Repeat the same using 2 different method calls.

Assignments HW:-

1. WAP to convert a HashSet to an array.
2. WAP to remove all of the elements from a HashSet.
3. WAP to add user defined objects of type Employee in a HashSet. Print the contents in the Set.
4. WAP to add user defined objects of type Employee in a HashSet using duplicate Employee object.
5. Print the contents in the Set.
6. WAP to add user defined objects of type Employee which now implements Comparable interface in a HashSet. Print the contents in the Set. Add duplicate Employee object and print the contents in the Set. Observe the values.
7. Do all above assignments in linkedhashset.
8. WAP to retrieve and remove the highest element of a TreeSet using a single method call. Repeat the same using 2 different method calls.
9. WAP to get the element in a TreeSet which is greater than or equal to the given element.
10. WAP to get the element in a TreeSet which is less than or equal to the given element.
11. WAP to get the element in a TreeSet which is strictly greater than or equal to the given element.
12. WAP to get an element in a TreeSet which is strictly less than the given element.
13. WAP to compare two TreeSets.
14. WAP to find the numbers less than 7 in a TreeSet which contains Integer objects.
15. WAP to find the numbers greater than 7 in a TreeSet which contains Integer objects.

Assignments(Additional):-

1. WAP to convert a HashSet to a List/ArrayList.

2. WAP to compare the elements in two HashSets.
3. WAP to compare two sets and retain elements which are same on both sets.
4. Do above assignments for linkedhashset.
5. WAP to add all the elements of a specified TreeSet to another TreeSet.
6. WAP to add user defined objects of type Employee in a TreeSet. Print the contents in the Set.
7. WAP to add user defined objects of type Employee in a TreeSet using duplicate Employee object.
8. Print the contents in the Set.
9. WAP to add user defined objects of type Employee which now implements Comparable interface in a TreeSet. Print the contents in the Set. Add duplicate Employee object and print the contents in the Set. Observe the values.
10. WAP to create a TreeSet and add user defined objects of type Employee (Employee is implementing Comparable interface in natural order). But use a Comparator which sorts in reverse order. Print the contents and observe the order of elements.

26. Queue, Stack

Assignments CW:-

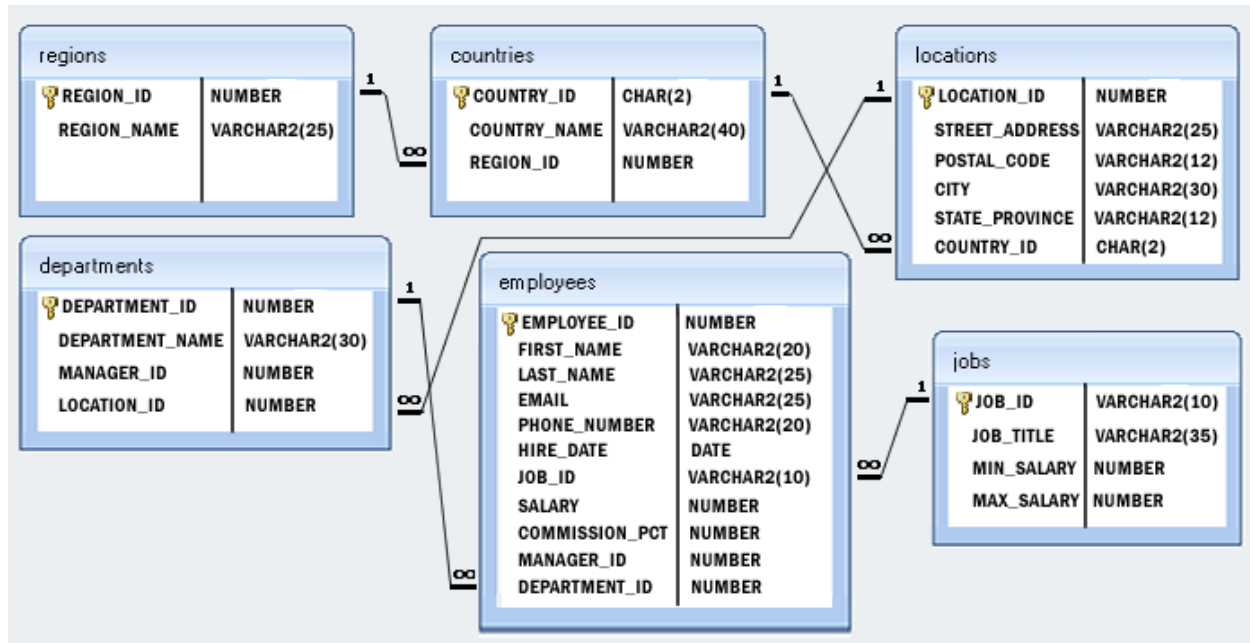
1. WAP to create a Queue with Integer objects without using generics
2. WAP to create a Queue with some colors (String) & using generics
3. WAP to create a Queue with user defined class objects & using generics
4. Use 2 different method calls to add elements to a queue.
5. WAP to check if a queue has values
6. WAP to create a Queue with Integer objects without using generics
7. WAP to create a Queue with some colors (String) & using generics
8. WAP to create a Queue using generics and add user defined class objects which does not implement comparable interface.
9. WAP to create a Queue using generics and add user defined class objects which implements comparable interface. Observe order when removing from queue.

Assignments HW:-

1. WAP to check the top element in a queue
2. WAP to remove an element from a queue
3. Use 2 different method calls to remove elements from a queue
4. WAP to attempt to remove non-existing elements from a queue
5. Use 2 different method calls to remove non-existing elements from a queue
6. WAP to implement your own implementation of Queue using an array internally
7. WAP to create a Queue using generics and add user defined class objects which does not implement comparable interface yet element are added in sorted in natural order. Observe order when removing from queue
8. WAP to create a Queue using generics and add user defined class objects in reverse order using comparator. Observe order when removing from queue.

MySQL

Consider the following DB structure:



27. Basic MySQL Part 1

Assignments CW :-

1. Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name"
2. Write a query to get unique department ID from employee table.
3. Write a query to get the total salaries payable to employees.
4. Write a query to get all employee details from the employee table order by first name, descending.
5. Write a query to get the names (first_name, last_name), salary, PF of all the employees (PF is calculated as 12% of salary).
6. Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary.
7. Write a query to get the maximum and minimum salary from employees table.

Assignments HW:-

1. Write a query to calculate $171 * 214 + 625$
2. Write a query to get the average salary and number of employees in the employees table
3. Write a query to get the number of employees working with the company
4. Write a query to get the number of jobs available in the employees table
5. Write a query get all first name from employees table in upper case
6. Write a query to get the first 3 characters of first name from employeeestable
7. Write a query to get the names (for example Ellen Abel, SundarAnde etc.) of all the employees from employees table
8. Write a query to get first name from employees table after removing white spaces from both side

Assignments Additional

1. Write a query to get the length of the employee names (first_name, last_name) from employees table
2. Write a query to check if the first_name fields of the employees table contains numbers.
3. Write a query to select first 10 records from a table.
4. Write a query to get monthly salary (round 2 decimal places) of each and every employee. Note: Assume the salary field provides the 'annual salary' information.

28. Basic MySQL Part 2

Assignments CW

1. Write a query to display the names (first_name, last_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000.
2. Write a query to display the names (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending alphabetical order by department ID.
3. Write a query to display the names (first_name, last_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000 and are in department 30 or 100.

Assignments HW

1. Write a query to display the names (first_name, last_name) and hire date for all employees who were hired in 1987.
2. Write a query to display the first_name of all employees who have both "b" and "c" in their first name.
3. Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary is not equal to \$4,500, \$10,000, or \$15,000.
4. Write a query to display the last names of employees whose names have exactly 6 characters.
5. Write a query to display the last names of employees having 'e' as the third character.

Assignments Additional

1. Write a query to display the jobs/designations available in the employees table.
2. Write a query to display the names (first_name, last_name), salary and PF (15% of salary) of all employees.
3. Write a query to select all record from employees where last name in 'BLAKE', 'SCOTT', 'KING' and 'FORD'.

29. Basic MySQL Part 3

Assignments CW

1. Write a query to list the number of jobs available in the employees table.
2. Write a query to get the total salaries payable to employees.
3. Write a query to get the minimum salary from employees table.
4. Write a query to get the maximum salary of an employee working as a programmer.

Assignments HW

1. Write a query to get the average salary and number of employees working the department 90.
2. Write a query to get the highest, lowest, sum, and average salary of all employees.
3. Write a query to get the number of employees with the same job.
4. Write a query to get the difference between the highest and lowest salaries.
5. Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.
6. Write a query to get the department ID and the total salary payable in each department.

Assignments Additional

1. Write a query to get the average salary for each job ID excluding programmer.
2. Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only.
3. Write a query to get the job ID and maximum salary of the employees where maximum salary is greater than or equal to \$4000.
4. Write a query to get the average salary for all departments employing more than 10 employees.

30. Intermediate MySQL Part 1

Assignments CW

1. Write a query to find the names (first_name, last_name) of the employees who are managers.
2. Write a query to find the names (first_name, last_name) and the salaries of the employees who have a higher salary than the employee whose last_name='Bull'.
3. Write a query to find the 5th maximum salary in the employees table.
4. Write a query to find the names (first_name, last_name) of the employees who have a manager and work for a department based in the United States. Hint : Write single-row and multiple-row subqueries
5. Write a query to fetch even numbered records from employees table.
6. Write a query to find the 4th minimum salary in the employees table.
7. Write a query to get nth max salaries of employees.

Assignments HW

1. Write a query to find the names (first_name, last_name) of all employees who works in the IT department.
2. Write a query to find the names (first_name, last_name), the salary of the employees whose salary is greater than the average salary.
3. Write a query to find the names (first_name, last_name), the salary of the employees whose salary is equal to the minimum salary for their job grade.
4. Write a query to find the names (first_name, last_name), the salary of the employees who earn more than the average salary and who works in any of the IT departments.
5. Write a query to find the names (first_name, last_name), the salary of the employees who earn more than Mr. Bull.
6. Write a query to get 3 minimum salaries.
7. Write a query to get nth min salaries of employees.

Assignments Additional

1. Write a query to find the names (first_name, last_name), the salary of the employees who earn the same salary as the minimum salary for all departments.
2. Write a query to find the names (first_name, last_name), the salary of the employees whose salary greater than the average salary of all departments.
3. Write a query to find the names (first_name, last_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = 'SH_CLERK'). Sort the results of the salary of the lowest to highest.
4. Write a query to find the names (first_name, last_name) of the employees who are not supervisors.
5. Write a query to display the employee ID, first name, last names, and department names of all employees.
6. Write a query to display the employee ID, first name, last names, salary of all employees whose salary is above average for their departments.
7. Write a query to select last 10 records from a table.
8. Write a query to list department number, name for all the departments in which there are no employees in the department.

9. Write a query to get 3 maximum salaries.

TQPP

31. Intermediate MySQL Part 2

Assignments CW:

1. Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments.
2. Write a query to find the names (first_name, last_name), department ID and name of all the employees.
3. Find the names (first_name, last_name), job, department number, and department name of the employees who work in London.
4. Write a query to find the employee id, name (last_name) along with their manager_id, manager name (last_name).
5. Find the names (first_name, last_name) and hire date of the employees who were hired after 'Jones'.

Assignments HW

1. Write a query to get the department name and number of employees in the department.
2. Find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90 from job history.
3. Write a query to display the department ID, department name and manager first name.
4. Write a query to display the department name, manager name, and city.

Assignments Additional

1. Write a query to display the job title and average salary of employees.
2. Display job title, employee name, and the difference between salary of the employee and minimum salary for the job.
3. Write a query to display the job history that were done by any employee who is currently drawing more than 10000 of salary
4. Write a query to display department name, name (first_name, last_name), hire date, salary of the manager for all managers whose experience is more than 15 years.

32. Basic and advanced JDBC Part 1

Assignments CW

1. WAP using JDBC to select and print all country names.
2. WAP using JDBC to select and print employee names along with their manager names.
3. WAP using JDBC to execute Stored Procedure using Callable Statement.
4. Write a program to execute update queries in a batch using commit and rollback.

Assignments HW

1. WAP using JDBC to delete employees if their salaries are equal to 0.
2. WAP using JDBC to transfer employees from manager with id 10 to manager with id 1. Delete employee with id 10. Do all using commit and rollback.

33. Basic and advanced JDBC Part 2

Assignments CW:

1. Write a console application where main menu would have names of tables from above database.
2. Read using jdbc, loop, insert using Collection ArrayList etc.
3. Choose the table to perform operations on
 - a. Regions
 - b. Countries
 - c. Locations
 - d. Departments
 - e. Employees
 - f. Jobs
 - g. Special Menu
4. Enter your choice:
5. Upon giving choice user would be shown following menu. Suppose user enters choice - 1
6. Get region name based on id
 - a. Add New Region
 - b. Delete Region
 - c. Update existing region
 - d. Go to main Menu
7. Same for Countries and so on.
8. Add additional special menu operations:
 - a. Show employees with highest salary
 - b. Show employees with lowest salary
 - c. List employees who start with name 'A'
 - d. Show employees who are hired in 2017
 - e. List department which has highest no of employees
 - f. Show top 3 list of managers who has maximum no of employees under them
 - g. Show minimum salary for each department
 - h. Show maximum salary for each department
 - i. Show locations which has maximum departments
 - j. Show alternate employee names under department with highest no of employees
 - k. Show list of region with highest no of employees