1. Write a Java program to get the character at the given index within the String

```
import java.util.Scanner;


public class Ques1 {
        public static void main(String[] args)
        {
                Scanner sc=new Scanner(System.in);
          String str ;
          System.out.println("Enter the String:");
          str=sc.nextLine();
          System.out.println("Original String = " + str);
          // Get the character at positions 0 and 10.
          int index1 = str.charAt(0);
          int index2 = str.charAt(10);

          // Print out the results.
          System.out.println("The character at position 0 is " +
            (char)index1);
          System.out.println("The character at position 10 is " +
            (char)index2);
        }
      }
```

2. Write a Java program to get the character (Unicode code point) at the given index within the String

```
import java.util.Scanner;


public class Ques2 {


  public static void main(String[] args) {
```

```
        Scanner sc=new Scanner(System.in);
   String str ;
   System.out.println("Enter the string:");
   str=sc.nextLine();
   System.out.println("Original String : " + str);

   // codepoint at index 1
   int val1 = str.codePointAt(2);

   // codepoint at index 9
   int val2 = str.codePointAt(3);

   // prints character at index1 in string
   System.out.println("Character(unicode point) = " + val1);
   // prints character at index9 in string
   System.out.println("Character(unicode point) = " + val2);
  }
}
```

3. Write a Java program to compare two strings lexicographically. Two strings are lexicographically equal if they are the same length and contain the same characters in the same positions

```
import java.util.Scanner;

public class Ques3 {
                public static void main(String[] args)
                  {
                     String str1,str2;
                     Scanner sc=new Scanner(System.in);
                     System.out.println("Input the two strings:");
                     str1=sc.nextLine();
                     str2=sc.nextLine();
                     System.out.println("String 1: " + str1);
```

```
        System.out.println("String 2: " + str2);

        // Compare the two strings.
        int result = str1.compareTo(str2);

        // Display the results of the comparison.
        if (result < 0)
        {
           System.out.println("\"" + str1 + "\"" +
              " is less than " +
              "\"" + str2 + "\"");
        }
        else if (result == 0)
        {
           System.out.println("\"" + str1 + "\"" +
              " is equal to " +
              "\"" + str2 + "\"");
        }
        else // if (result > 0)
        {
           System.out.println("\"" + str1 + "\"" +
              " is greater than " +
              "\"" + str2 + "\"");
        }
     }
  }
```

4. Write a Java program to counts occurrences of a certain character in a given string

```
import java.util.Scanner;

public class Ques4 {
```

```
public static void main(String args[])
                    {

                    String input;
                    Scanner sc=new Scanner(System.in);
                    System.out.println("Enter the string:");
                    input = sc.nextLine();

                    char search ;
                    System.out.println("Enter the character to search:");
                    search = sc.next().charAt(0);// Character to search is 'a'.

                    int count=0;
                    for(int i=0; i<input.length(); i++)
                    {
                       if(input.charAt(i) == search)
                       count++;
                    }

                    System.out.println("The Character '"+search+"' appears "+count+" times.");
                    }}
```

5. Write a Java program to concatenate a given string with itself of a given number of times.

```
import java.util.Scanner;

public class Ques5 {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
    String str,s1 =" ";
    Scanner sc=new Scanner(System.in);
```

```
System.out.println("Enter the string to conactenate");
str=sc.nextLine();
int n;
System.out.println("Enter the number of times to concatenate the given string");
n=sc.nextInt();
for(int i=0;i<n;i++)
{
      s1+=str;
}  System.out.println(s1);
}
      }
```

6.check the given string is panlidrome or not

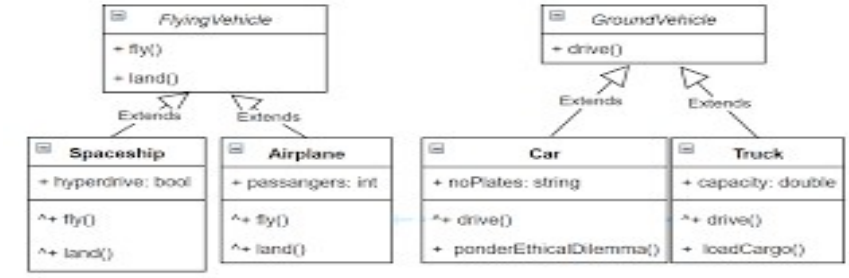
```
import java.util.Scanner;

public class Ques7 {

      public static void main(String[] args) {
            // TODO Auto-generated method stu
              String x, y = "";
             Scanner a = new Scanner(System.in);
             System.out.print("Enter  string you want to check:");
            x = a.nextLine();
              int l = x.length();
             for(int k = l - 1; k >= 0; k--)
            {
                y = y + x.charAt(k);
             }
             if(x.equalsIgnoreCase(y))
              {
```

```
                System.out.println("The string is palindrome.");
            }
            else
            {
                System. out.println("The string is not a palindrome.");
            }
        }
    }
```

8.Java Program to prove that strings are immutable in java

```
public class Ques8
    {
        public static void main(String[] args)
        {
            String s1 = "JAVA";

            String s2 = "JAVA";

            System.out.println(s1 == s2);
            //Output : true
            System.out.println("s1 and s2 are equal");
            s1 = s1 + "course";

            System.out.println(s1 == s2);        //Output : false
            System.out.println("s1 and s2 are not equal");
        }

    }
```

6. Java program to implement below classes using inheritance




```
package Inheritancepack9;


public class FlyingVehicle {
    public void fly() {
            System.out.println("Fly method of flying vehicle class");
    }
    public void land() {
            System.out.println("land method of flying vehicle class");
}
package Inheritancepack9;


public class GroundVehicle {
  public void drive(){
  System.out.println("drive method of GroundVehicle");
  }
}
package Inheritancepack9;


public class Spaceship extends FlyingVehicle {
  boolean hypendrive;


  public Spaceship(boolean hypendrive) {
          super();
          this.hypendrive = hypendrive;
}
```

```
  @Override
    public void fly() {
                super.fly();
                System.out.println("Fly method of spaceship class");
 }
 public void land() {
               super.land();
                System.out.println("land method of spaceship class");}}
```

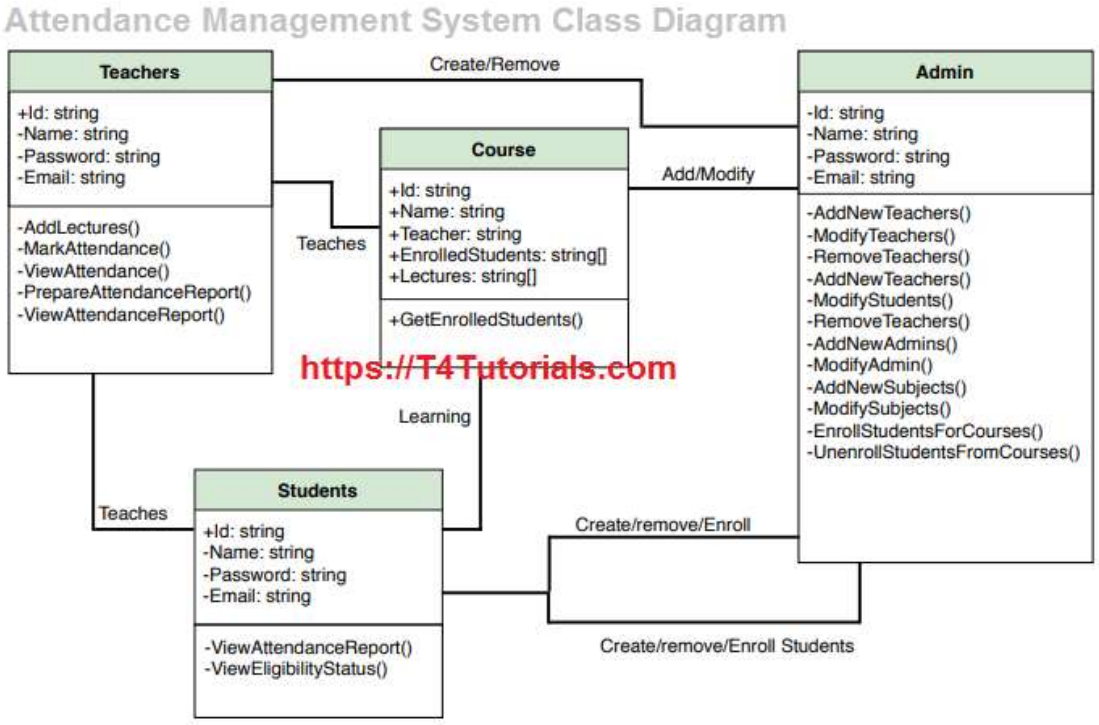
```
package Inheritancepack9;

public class Airplane extends FlyingVehicle{
    int passengers;

            public Airplane(int passengers) {
                        super();
                        this.passengers = passengers;
            }
             @Override
            public void fly() {

              System.out.println("Fly method of airpalne class");
 }
 public void land() {

              System.out.println("land method of airplane class");
 }

}
```

```
package Inheritancepack9;
public class Truck extends GroundVehicle {
	double capacity;
public Truck() {

}

public Truck(double capacity) {
	super();
	this.capacity = capacity;
	System.out.println("capacity is "+capacity);
}
@Override
public void drive() {
	// TODO Auto-generated method stub

	System.out.println("drive method of truck class");
}
public void loadCargo() {
	System.out.println("loadcargo method of truck class");
}
}
package Inheritancepack9;

public class Car extends GroundVehicle{
	 String noPlates;
	 public Car() {

	 }

	 public Car(String noPlates) {
```

```
		super();
		this.noPlates = noPlates;
		System.out.println("noplate "+ noPlates);
	}

	@Override
	public void drive() {
		// TODO Auto-generated method stub
		super.drive();
		System.out.println("drive method of car class");

	}
	public void pounderEthicalDlemma() {
		 System.out.println("pounderEthicalDlemma method of car class");
	}

	}
package Inheritancepack9;

public class Test {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

    Spaceship ss=new Spaceship(true);
		ss.fly();
		ss.land();
		Airplane a=new Airplane(32);
		int passengers=100;
		Airplane a1=new Airplane(passengers);
		a1.fly();
```

```
        a1.land();
        Car c=new Car("ka passing");
        c.drive();
        c.pounderEthicalDlemma();

        Truck t=new Truck(1234.5);

        t.drive();
        t.loadCargo();

    }}
```

7. Write a java program to implement the below diagram



ADMIN:

```
package attendancemanagement;

public class Admin extends Person {
```

```
Teachers[] teacherList=new Teachers[15];
	static int count=0;



	public Admin(String id, String name, String password, String email) {
		super(id, name, password, email);
		// TODO Auto-generated constructor stub
	}

public void addNewTeacher(Teachers teacher)
{
	teacherList[count++]=teacher;

}
public void viewTeacherList()
{
	for(int i =0;i<count;i++)
	{
		System.out.println("teacher list  : "+teacherList[i]);
	}
}

public void modifyTeacherInfo(String id,String password)
{
	for(int i=0;i<count;i++)
	{
		if(teacherList[i].getId().equals(id))
		{
			teacherList[i].setPassword(password);
			break;
		}}}
```

```
public void removeTeacherById(String id)
{   int pos=-1;
        for(int i=0;i<count;i++)
        {
                if(teacherList[i].getId().equals(id))
                {
                        pos= i;
                        break;
                }
        }
        for(int i=pos;i<count;i++)
        {
                teacherList[i] = teacherList[i+1];

        }
        if(pos>=0)
        {
                count--;
        }
        }
public void viewTeacherById(String id)
{
        for(int i =0;i<count;i++)
        {
                if(teacherList[i].getId().equals(id))
                {
                        System.out.println("Teacher Details  :  "+teacherList[i]);
                }
        }

        }
```

```
Student[] studentList=new Student[15];
static int count1=0;



/*public Admin(String id, String name, String password, String email) {
	super(id, name, password, email);
	// TODO Auto-generated constructor stub
}1111*/

public void addNewStudent(Student student)
{
studentList[count1++]=student;

}
public void viewStudentList()
{
for(int i =0;i<count1;i++)
{
	System.out.println("student list  : "+studentList[i]);
}
}

public void modifyStudentInfo(String id,String password)
{
for(int i=0;i<count1;i++)
{
	if(studentList[i].getId().equals(id))
	{
		studentList[i].setPassword(password);
		break;
	}}}
```

```
public void removeStudentById(String id)
{   int pos=-1;
for(int i=0;i<count1;i++)
{
        if(studentList[i].getId().equals(id))
        {
                pos= i;
                break;
        }
}
for(int i=pos;i<count1;i++)
{
        studentList[i] = studentList[i+1];

}
if(pos>=0)
{
        count--;
}
}
public void viewStudentById(String id)
{
for(int i =0;i<count1;i++)
{
        if(studentList[i].getId().equals(id))
        {
                System.out.println("student Details  :  "+studentList[i]);
        }}}}
```

Teachers:

```
package attendancemanagement;
```

```
public class Teachers extends Person {

		public Teachers(String id, String name, String password, String email) {
			super(id, name, password, email);
			// TODO Auto-generated constructor stub
		}}
```

Students:

```
package attendancemanagement;

public class Student extends Person{

		public Student(String id, String name, String password, String email) {
			super(id, name, password, email);
			// TODO Auto-generated constructor stub
	}
	}
```

```
package attendancemanagement;

public class Course extends Student{

	public  Course(String id, String name, String Teacher, String[] EnrolledStudents,String[]
Lectures) {
		super(id, name, Teacher, EnrolledStudents,Lectures);
		// TODO Auto-generated constructor
	}



}
```

TestAttendance:

```
package attendancemanagement;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class TestAttendance {

    public static void main(String[] args) throws IOException{
        // TODO Auto-generated method stub
        char ch1;
        do
        {
        System.out.println("1 for Admin  ");
        System.out.println("2 for Student  ");
        System.out.println("3 for teacher ");
        System.out.println("Enter option 1/2/3  ");
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        int op = Integer.parseInt(bufferedReader.readLine());
        switch(op)
        {
        case 1:
            Admin admin = new Admin("1233", "Durgesh", "asdfgh", "durgesh@gmail.com") ;
            char ch;
            do {
      System.out.println("1 for Add teacher  ");
        System.out.println("2 for ViewTeacherList  ");
        System.out.println("3 for Modify Teacher Details ");
        System.out.println("4 for Delete Teacher Details ")
```

```
System.out.println("5 for Show  Teacher Details  By Id ");
                    System.out.println("Enter option 1/2/3/4/5 ");
                    int op1 = Integer.parseInt(bufferedReader.readLine());
                      switch(op1)
                      {
                      case 1:
                        System.out.println("Enter teacher's id name password and email ");
                        Teachers teachers = new Teachers(bufferedReader.readLine(),
bufferedReader.readLine(),bufferedReader.readLine(), bufferedReader.readLine());


                       admin.addNewTeacher(teachers);
                      break;
                      case 2:
                        admin.viewTeacherList();


                      break;
                     case 3:
                        System.out.println("Enter Exsisting teacher Id And Password");
                        admin.modifyTeacherInfo(bufferedReader.readLine(),bufferedReader.readLine());
                      break;
                       case 4:
                        System.out.println("Enter Exsisting teacher Id To Delete Teacher Information");
                        admin.removeTeacherById(bufferedReader.readLine());
                      break;
                      case 5:
                        System.out.println("Enter Exsisting teacher Id ");
                        admin.viewTeacherById(bufferedReader.readLine());
                        break;
                      default: System.out.println("Invalid Option");  }
                      System.out.println("Do you want to continue");
                      ch = bufferedReader.readLine().charAt(0);
```

```
}while(ch=='y'|| ch =='Y');
                    break;
                case 2:

                    Admin admin1 = new Admin("1233", "Durgesh", "asdfgh", "durgesh@gmail.com") ;
                    char ch11 = 0;
                    do {
            System.out.println("1 for Add student  ");
                System.out.println("2 for ViewStudentList  ");
                System.out.println("3 for Modify Student Details ");
                System.out.println("4 for Delete Student Details ");

                System.out.println("5 for Show  Student Details  By Id ");
                System.out.println("Enter option 1/2/3/4/5 ");
                int op1 = Integer.parseInt(bufferedReader.readLine());
                  switch(op1)
                  {
                  case 1:
                    System.out.println("Enter student's id name password and email ");
                    Student students = new Student(bufferedReader.readLine(),
bufferedReader.readLine(),bufferedReader.readLine(), bufferedReader.readLine());

                   admin1.addNewStudent(students);
                  break;
                  case 2:
                    admin1.viewStudentList();

                  break;
                 case 3:
                    System.out.println("Enter Exsisting student Id And Password");
                    admin1.modifyStudentInfo(bufferedReader.readLine(),bufferedReader.readLine());
```

```
            break;
             case 4:
              System.out.println("Enter Exsisting student Id To Delete Student Information");
              admin1.removeStudentById(bufferedReader.readLine());
            break;
            case 5:
              System.out.println("Enter Exsisting student Id ");
              admin1.viewStudentById(bufferedReader.readLine());
              break;
            default: System.out.println("Invalid Option");
          }
            System.out.println("Do you want to continue");
            ch = bufferedReader.readLine().charAt(0);
            }while(ch11=='y'|| ch11=='Y');
              break;   case 3:
              break;
      default :
              System.out.println("Enter Valid Option ");
      }
      System.out.println("Do you want to continue");
      ch1 = bufferedReader.readLine().charAt(0);
                    }while(ch1=='y'|| ch1 =='Y');
            }

}
```