Part A

1. Create a Java class called Student with the following details as variables within it.
i) USN
(ii) Name
(iii) Branch
(iv) Phone
create n Student objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.

```
class Student
{
            String usn;
            String name;
            String branch;
            String phone;

        public String getUsn() {
        return usn;
           }

        public void setUsn(String usn)
        {
        this.usn = usn;
        }

        public String getName()
        {
        return name;
        }

        public void setName(String name)
        {
              this.name = name;
        }

        public String getBranch()
        {
```

```
return branch;
}

public void setBranch(String branch)
{
this.branch = branch;
}

public String getPhone()
{
return phone;
}

public void setPhone(String phone)
{
this.phone = phone;
}


public static void main(String[] args)
{
System.out.print("Enter the number of Student: ");
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();

Student[] studentList = new Student[n];
for (int i = 0; i< n; i++)
studentList[i] = new Student();

System.out.println("Enter the Student Details");
for (int i = 0; i< n; i++) {
System.out.println("Enter the Student "+(i+1)+" Details");
System.out.print("Enter the usn: ");
studentList[i].setUsn(scn.next());
System.out.print("Enter the Name: ");
studentList[i].setName(scn.next());
System.out.print("Enter the Branch: ");
studentList[i].setBranch(scn.next());
System.out.print("Enter the Phone Number: ");
studentList[i].setPhone(scn.next());
```

```
            }

        System.out.println();
        System.out.println("The Student Details are");
        for (int i = 0; i< n; i++) {
        System.out.println("The Student "+(i+1)+" Details");
        System.out.println("Student Usn: " + studentList[i].getUsn());
        System.out.println("Student Name: " + studentList[i].getName());
        System.out.println("Student Branch: " + studentList[i].getBranch());
        System.out.println("Student Phone Number: " +   studentList[i].getPhone());
        System.out.println();
         }
  }
}
```

OUTPUT:

```
Enter the number of Student: 2
Enter the Student Details
Enter the Student 1 Details
Enter the usn: 101
Enter the Name: shalini
Enter the Branch: cse
Enter the Phone Number: 9663101601
Enter the Student 2 Details
Enter the usn: 100
Enter the Name: shyla
Enter the Branch: cse
Enter the Phone Number: 9972653309

The Student Details are
The Student 1 Details
Student Usn: 101
Student Name: shalini
Student Branch: cse
Student Phone Number: 9663101601

The Student 2 Details
Student Usn: 100
Student Name: saumya
```

Student Branch: cse
Student Phone Number: 9972653309

2. Write a Java programs to demonstrate working of polymorphism in Java:

a. Method Overloading

b. Constructor Overloading.

```
class MethodOverloading {
   private static void display(int a){
      System.out.println("Got integer data with one argument");
   }

   private static void display(int a, int b){
      System.out.println("Got integer data with two arguments");
   }

   private static void display(String a){
      System.out.println("Got String object.");
   }
   public static void main(String[] args) {
      display(1);
      display(1, 4);
      display("Hello");
   }
}
```

Output:

```
Got integer data with one argument
Got integer data with two arguments
Got String object.
```

```
public class ConstructorOverloading {
   int id;
   String name;

   ConstructorOverloading(){
   System.out.println("this a default constructor");
   }
```

```
ConstructorOverloading(int i, String n){
id = i;
name = n;
}

public static void main(String[] args) {
ConstructorOverloading s = new ConstructorOverloading();
System.out.println("\nDefault Constructor values: \n");
System.out.println("Student Id : "+s.id + "\nStudent Name : "+s.name);

System.out.println("\nParameterized Constructor values: \n");
ConstructorOverloading student = new ConstructorOverloading(10, "Shalini");
System.out.println("Student Id : "+student.id + "\nStudent Name : "+student.name);
}
}
```

Output:

this a default constructor

Default Constructor values:

Student Id : 0
Student Name : null

Parameterized Constructor values:

Student Id : 10
Student Name : Shalini

3. Write a Java Programs to demonstrate the usage of the following:

- **a. Control structures of Java**
- **b. Looping structures of Java**
- **c. Break and Continue statements**

```
public class ForEx {

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

```
{
int n = 3;

for (int i = 0; i< n; i++)
  {
   System.out.println("Tick " + i);
  }
}
}
```

OUTPUT:

```
Tick 0
Tick 1
Tick 2
```

```
public class Dowhile {

public static void main(String[] args)
{
int n = 4;
do
  {
System.out.println("Tick " + n);
n--;
  }
while(n>0);
}
}
```

OUTPUT:

```
Tick 0
Tick 1
Tick 2
```

```
public class SwithEx {
public static void main (String args[])
  {
int rc = Integer.parseInt(args[0]);

String msg;
```

```
switch (rc) {
case 1:
msg = "Syntax error";
break;
case 2:
msg = "Undefined variable";
break;
default:
msg = "Unknown error";
break;
        }
System.out.println("\n" + msg);
}
}
```

OUTPUT:

Undefined variable

```
public class Break
{
        public static void main (String[] argv)
        {
                for (int i=0; i<=15; i++)
                {
                System.out.println("\n" +i);
                if ( (i&2) == 0 )
                continue;
                if ( (i%2) != 0)
                break;
                System.out.println("\nThat's odd");
                }
        }

}
```

OUTPUT:

0

1

2

That's odd

3

4. Write a Java programs to demonstrate the usage of the For-Each Style for-loop- statement

```
public class ForEach
{
        public static void main(String args[])
        {
        int array1[] = { 10, 20, 30 };
        for(int ele1: array1)
        System.out.println("1D Array elements "+ele1+"\t");


        int[][] contents = { { 88, 66, 79 }, { 56, 25, 39 }, { 58, 47, 69 } };

   System.out.println("Loop Using Enhanced for loop:");
   for (int[] eachRow : contents)
    {
         for (int j : eachRow)
         {
               System.out.print(j + "\t");
         }
         System.out.println("");
    }
}
}
```

OUTPUT:

1D Array elements 10
1D Array elements 20
1D Array elements 30

Loop Using Enhanced for loop:
88 66 79
56 25 39
58 47 69

5. Write a program in java to generate m x n multiplication table where in m and n values are obtained as command line arguments.

```
import java.util.*;
public class MultiplicationTable {

    public static void main(String[] args) {
        int m=Integer.parseInt(args[0]);
        int n=Integer.parseInt(args[1]);
        for(int i = 1; i <= n; ++i)
        {
            System.out.printf("%d * %d = %d \n", m, i, m * i);
        }
    }
}
```

OUTPUT:

```
java MultiplicationTable.java 5 10
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

6. Write a program in java to accept and display employee id, employee name and salary through keyboard and StringTokenizer object.

```
class Employee
```

```
{
        private String employeeId;
        private String name;
        private int salary;

        public String getEmployeeId()
        {
        return employeeId;
        }

        public void setEmployeeId(String employeeId)
        {
        this.employeeId = employeeId;
         }

        public String getName()
        {
        return name;
        }

        public void setName(String name)
        {
              this.name = name;
        }

        public int getSalary()
        {
        return salary;
        }

        public void setSalary(int salary)
        {
        this.salary = salary;
        }

                public static void main(String[] args)
                {
                System.out.println("Enter the employee details in format given");
                System.out.println("EmployeeId Name Salary");
```

```
                Scanner scn = new Scanner(System.in);
                String txt = scn.nextLine();
                Employee emp = new Employee();

            StringTokenizer tokenTxt = new StringTokenizer(txt, " ");

            emp.setEmployeeId(tokenTxt.nextToken());
            emp.setName(tokenTxt.nextToken());
            emp.setSalary(Integer.parseInt(tokenTxt.nextToken()));

            System.out.println("Employee Details are: ");
            System.out.println("Employee Id: " + emp.getEmployeeId());
            System.out.println("Employee Name: " + emp.getName());
            System.out.println("Employee Salary: " + emp.getSalary());
}
}
```

OUTPUT:

```
Enter the employee details in format given
EmployeeId Name Salary
123 Raja 12500
Employee Details are:
Employee Id: 123
Employee Name: Raja
Employee Salary: 12500
```

7. Write a Java program to facilitate the multilevel inheritance. (Also demonstrate the use of constructor in multilevel inheritance)

```
class travel
{
        travel()
        {
                System.out.println("Travel constructor called");
        }
}
class Indonesia extends travel
{
        Indonesia()
```

```
	{
		System.out.println("Indonesia constructor called");
	}
}
class Bali extends Indonesia
{
	Bali()
	{
		System.out.println("Bali constructor called");
	}
}
public class MultilevelInheritance
{
	public static void main()
	{
		System.out.println("Order of execution is ");
		Bali b=new Bali();
	}
}
```

Output:

```
Order of execution is
Travel constructor called
Indonesia constructor called
Bali constructor called
```

8. Write a Java Program to demonstrate the concept of hierarchical inheritance. (implement this program with super keyword, final keyword, constructor and method overriding concept)

```
class Employee{
   final float salary = 40000;
   Employee(){System.out.println("Employee constructor called");}
   void yearsOfExp()
   {
      int Exp=25;
      System.out.println("Years of experience for an Employee is:"+Exp);
   }
   }
   class PermanentEmp extends Employee{
```

```
double hike = 0.5;
PermanentEmp(){System.out.println("PermanentEmp constructor called");}
void yearsOfExp()
{
   int Exp=35;
   System.out.println("Years of experience for Permanent Employee is:"+Exp);
}
}
class TemporaryEmp extends Employee{
double hike = 0.35;
TemporaryEmp(){System.out.println("TemporaryEmp constructor called");}
void yearsOfExp()
{
   int Exp=15;
   System.out.println("Years of experience for Temporary Employee is:"+Exp);
}
}
public class hier101
{
public static void main(String args[]){
PermanentEmp p = new PermanentEmp();
TemporaryEmp t = new TemporaryEmp();
System.out.println("Salary for all Employees is :" +p.salary);
p.yearsOfExp();
System.out.println("Hike for Permanent Employee is:" +p.hike);
t.yearsOfExp();
System.out.println("Hike for Temporary Employee is :" +t.hike);
}
}
```

Output:

Employee constructor called
PermanentEmp constructor called
Employee constructor called
TemporaryEmp constructor called
Salary for all Employees is 40000
Years of experience of Permanent Employees is 35
Hike for permanent employee is 0.5
Years of experience of Temporary Employee is 15
Hike for temporary employee is 0.35

9. Write a java program to demonstrate dynamic method dispatch and abstract keyword with class and methods.

```
abstract class Bank{
abstract float getRateOfInterest(){}
}
class SBI extends Bank{
float getRateOfInterest(){return 8.4f;}
}
class ICICIextends Bank{
float getRateOfInterest(){return 7.8f;}
}

class AXIS extends Bank{
float getRateOfInterest(){return 9.7f;}
}
class dynabs101{
public static void main(String args[]){
Bank b;
b=new SBI();
System.out.println("SBI Rate of Interest: "+b.getRateOfInterest());
b=new ICICI();
System.out.println("ICICI Rate of Interest: "+b.getRateOfInterest());
b=new AXIS();
System.out.println("AXIS Rate of Interest: "+b.getRateOfInterest());
}
}
```

Output:

SBI Rate of Interest 8.4
ICICI Rate of Interest 7.8
AXIS Rate of Interest 9.7

10. Try to implement the concept of multiple inheritance in Java with the use of interface.

```
interface IEat {
   void eat();
```

```
}
interface ITravel {
  void travel();
}
class Me implements IEat, ITravel {
  public void eat() {
    System.out.println("I am eating");
  }
  public void travel() {
    System.out.println("I am traveling");
  }
}
public class multipleinherit101 {
  public static void main(String args[]) {
    Me m = new Me();
    m.eat();
    m.travel();
  }
}
```

Output:

```
I am eating
I am traveling
```

11. Write a different java program for generating following types of exception

a. NullPointerException
b. ArrayIndexOutOfBoundException
c. ArithmeticException
d. NumberFormatException
e. StringIndexOutOfBoundException

```
        import java.util.*;
class ExceptionTypes {
    public static void main(String[] args) {
        try {
            int a=3/0;
        }
```

```
        catch(ArithmeticException e) {
            System.out.println(e);
        }
        System.out.println("Arithmetic Exception");
        try {
            String str=null;
            System.out.println(str.length());
        }
        catch(NullPointerException e) {
            System.out.println(e);
        }
        System.out.println("Null Pointer Exception");
        try {
            int a[]=new int[3];
            a[4]=15;
        }
        catch(ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        }
        System.out.println("Array Index Out Of Bounds Exception");
        try {
            String s="Shalini";
            char ch=s.charAt(25);
            System.out.println(s);
            System.out.println(ch);
        }
        catch(StringIndexOutOfBoundsException e) {
            System.out.println(e);
        }
        System.out.println("String Index Out Of Bounds Exception");
        try {
            String w="Shalini";
            int n=Integer.parseInt(w);
        }
        catch(NumberFormatException e) {
            System.out.println(e);
        }
        System.out.println("Number Format Exception");
    }
}
```

Output:

```
java.lang.ArithmeticException: / by zero
Arithmetic Exception
java.lang.NullPointerException
Null Pointer Exception
java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 3
Array Index Out Of Bounds Exception
java.lang.StringIndexOutOfBoundsException: String index out of range: 25
String Index Out Of Bounds Exception
java.lang.NumberFormatException: For input string: "Shalini"
Number Format Exception
```

12. a) Write a program to demonstrate user defined exception.

```
import java.util.*;

class UserDefinedException extends Exception{
	public static void main(String[] args) {
		String s="Shalini";
		try {
			if(s.length()<20)
				throw new Exception("String length should be more than 20.");
		}
		catch(Exception e) {
			System.out.println("Exception detected");
			System.out.println(e.getMessage());
		}
	}
}
```

Output:

```
Exception detected
String length should be more than 20.
```

12. b) Write program in java to demonstrate the use of throw and throws in exception.

```
import java.io.IOException;
```

```
class ThrowThrows{
	void fun()throws IOException{
		throw new IOException("Device Error");
	}
}
public class ThrowAndThrows {
	public static void main(String[] args) {
		try {
			ThrowThrows obj=new ThrowThrows();
			obj.fun();
		}
		catch(Exception e) {
			System.out.println("Exception handled");
		}
		System.out.println("Other statements");
	}
}
```

Output:
Exception handled
Other statements

13. Write a program to create a new thread by extending a thread class.

a. Get the current thread name
b. Set the highest priority to the newly created thread
c. Pause a thread for 2.5 seconds.
d. Check whether the thread is in running state or not.
e. Verify your newly created thread must be completed first before your main thread is completed.

```
import java.util.*;

class NewThread extends Thread{
	public void run() {
		System.out.println("Thread "+Thread.currentThread().getName()+" is
running");
	}
```

```
	public static void main(String[] args) {
		NewThread t1=new NewThread();
		t1.start();
		System.out.println("Current thread name is
"+Thread.currentThread().getName());
		t1.setPriority(MIN_PRIORITY);
		System.out.println("Thread priority is "+t1.getPriority());
		try {
			Thread.sleep(2500);
		}catch(InterruptedException e) {
			System.out.println(e);
		}
		System.out.println("New thread state is "+t1.getState());
		System.out.println("Main thread state is "+Thread.currentThread().getState());
	}
}
```

Output:

```
Thread priority is 1
Thread Thread-0 is running
New thread state is TERMINATED
Main thread state is RUNNABLE
```

14. Assume that only one copy of the book is available in Amazon and four customers are trying to place the order for book at the same time. Write a java program using threads which prints book confirmed for one person and “out of stock” for others.

```
public class Amazon implements Runnable{
	static int noOfBooks=1;
	synchronized public void run() {
		if(noOfBooks>0) {
			noOfBooks--;
			System.out.println("Confirmed");
		}
		else {
			System.out.println("Out of stock");
		}
	}
```

```
	public static void main(String[] args) {
		Thread t1=new Thread(new Amazon(),"Thread1");
		Thread t2=new Thread(new Amazon(),"Thread2");
		Thread t3=new Thread(new Amazon(),"Thread3");
		Thread t4=new Thread(new Amazon(),"Thread4");
		t1.start();
		t2.start();
		t3.start();
		t4.start();

	}
}
```

Output:

```
Confirmed
Out of stock
Out of stock
Out of stock
```

15. Write a program for creating three threads randomly using following methods:

a. By extending Thread class
b. By implementing Runnable interface

a.

```
import java.util.*;

class ThreeThreads extends Thread{
	ThreeThreads(String threadName) {
		super(threadName);
	}
	public void run() {
		System.out.println("Thread "+Thread.currentThread().getName()+" is
runnning");
	}
	public static void main(String[] args) {
		ThreeThreads t1=new ThreeThreads("t1");
		ThreeThreads t2=new ThreeThreads("t2");
		ThreeThreads t3=new ThreeThreads("t3");
```

```
            t1.start();
            t2.start();
            t3.start();
        }
}
```

Output:

```
Thread t1 is running
Thread t2 is running
Thread t3 is running
```

b.

```
import java.io.*;
class threeb implements Runnable
{
        public void run()
        {
                System.out.println("Thread is running inside");
        }
        public static void main(String args[])
        {
                threeb th1=new threeb();
                Thread t1=new Thread(th1);
                t1.start();
                Thread t2=new Thread(th1);
                t2.start();
                Thread t3=new Thread(th1);
                t3.start();
        }
}
```

Output:

```
Thread is running inside
Thread is running inside
Thread is running inside
```

16. Write a Java program that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, Absolute path of file and the length of the file in bytes?

```
import java.io.File;
```

```
public class FileInfo{
	public static void main(String[] args){
		File obj=new File(args[0]);
		if(obj.exists()){
			System.out.println("Name of file is "+obj.getName());
			System.out.println("Absolute path is "+obj.getAbsolutePath());
			System.out.println("Can we read the file "+obj.canRead());
			System.out.println("Can we write the file "+obj.canWrite());
			System.out.println("Length of the file "+obj.length());
		}
	}
}
```

Output:

```
> java FileInfo sample.txt
Name of the file is sample.txt
Absolute path is C:\Users\Shalini\Documents\Programs\sample.txt
Can we read the file true
Can we write the file true
Length of the file 18
```

17. Write a program in java to read filename from user, read data from file using File Reader and improve its efficiency and display the contents of the file.

```
import java.io.*;

public class practice {

   public static void main(String[] args) throws Exception
   {
      FileReader file = new FileReader(args[0]);

      BufferedReader br = new BufferedReader(file);

      String st;
      while ((st = br.readLine()) != null)
         System.out.println(st);
   }
}
```

Output:
>java Readwrite sample.txt
I'm Shalini
I love music
I love traveling and exploring new places

Part B

1. Write a program that implements the client/server application. The client sends the data to the server; the server receives the data, uses it to produce a result and then sends the result back to the client. The client displays the result on the console.

server.java

```
import java.net.*;
import java.io.*;
class server{
   public static void main(String args[])throws Exception{
      ServerSocket ss=new ServerSocket(3333);
      Socket s=ss.accept();
      DataInputStream din=new DataInputStream(s.getInputStream());
      DataOutputStream dout=new DataOutputStream(s.getOutputStream());
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

      String str="",str2="";
      while(!str.equals("stop")){
         str=din.readUTF();
         System.out.println("client says: "+str);
         str2=br.readLine();
         dout.writeUTF(str2);
         dout.flush();
      }
      din.close();
      s.close();
      ss.close();
   }
}
```

client.java

```
import java.net.*;
```

```
import java.io.*;
class client{
   public static void main(String args[])throws Exception{
      Socket s=new Socket("localhost",3333);
      DataInputStream din=new DataInputStream(s.getInputStream());
      DataOutputStream dout=new DataOutputStream(s.getOutputStream());
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

      String str="",str2="";
      while(!str.equals("stop")){
         str=br.readLine();
         dout.writeUTF(str);
         dout.flush();
         str2=din.readUTF();
         System.out.println("Server says: "+str2);
      }

      dout.close();
      s.close();
   }
}
```

Output:

```
Client- Hi Server
Server- Client says: Hi Server
Server- Hello Client ! What can I do for you ?
Client- Server says: Hello Client ! What can I do for you ?
Client- Can you send me the result of 2^10
Server- Client says: Can you send me the result of 2^10
Server- Yes, after computation, the result is 1024
Client- Server says: Yes, after computation, the result is 1024
Client- Okay thanks !
Server- Client says: Okay thanks !
Server- You're welcome
Client- Server says: You're welcome
Client- stop
Server- Client says: stop
```

2. Write java program to demonstrate the following String functions

a. length()

b. isEmpty(), isBlank()
c. charAt()
d. equals(),equalsIgnoreCase()
e. compareTo() and compareToIgnoreCase()
f. startsWith() and endsWith()
g. substring()
h. concat()
i. replace(), replaceFirst(), and replaceAll()
j. contains()
k. reverse();
l. split()
m. join()
n. toLowerCase() and toUpperCase()
o. trim( )

```
public class methods {
    public static void main(String[] args) {
        String s1 = "Hello Java";
        System.out.println("The length of string is " + s1.length());
        System.out.println(s1.charAt(2));
        // System.out.println(s1.charAt(20));
        String emptyStr = "";
        String whitespacesStr = "  ";
        System.out.println("1:"+emptyStr.isEmpty());
        System.out.println("2:"+whitespacesStr.isEmpty());
        System.out.println("3:"+emptyStr.isBlank());
        System.out.println("4:"+whitespacesStr.isBlank());
        System.out.println("5:"+"Java".equals("java"));
        System.out.println("6:"+"Java".equalsIgnoreCase("java"));
        System.out.println("7:"+"Java".compareTo("java"));
        System.out.println("8:"+"Java".compareToIgnoreCase("JAVA"));
        System.out.println("9:"+"Java".startsWith("J"));
        System.out.println("10:"+"Java".endsWith("a"));
        System.out.println("11:"+"Hello Shalini".substring(5));
        System.out.println("12:"+"Hello Shalini".substring(5,10));
        System.out.println("13:"+"Hello ".concat("Shalini"));
        System.out.println("14:"+"Hello Shalini".replace("Sh","M"));
        System.out.println("15:"+"Hello Shalini".replaceFirst("l","m"));
        System.out.println("16:"+"Hello Shalini".replaceAll("l","m"));
        System.out.println("17:"+"Shalini".contains("Sh"));
```

```
        System.out.println("18:"+"Shalini".split("a"));
        System.out.println("19:"+" ".join(",","A","B"));
        System.out.println("20:"+"        Shalini      ".trim());
        System.out.println("21:"+"Shalini".toUpperCase());
        System.out.println("22:"+"SHAlini".toLowerCase());
    }
}
```

Output:

```
The length of string is 10
l
1:true
2:false
3:true
4:true
5:false
6:true
7:-32
8:0
9:true
10:true
11: Shalini
12: Shal
13:Hello Shalini
14:Hello Malini
15:Hemlo Shalini
16:Hemmo Shamini
17:true
18:[Ljava.lang.String;@5d3411d
19:A,B
20:Shalini
21:SHALINI
22:shalini
```

3. Demonstrate the different types of selectors in CSS.
CSS Element Selector
CSS Id Selector
CSS Class Selector
CSS Universal Selector
CSS Group Selector

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>selectors</title>
    <style>
        p {
            text-align: center;
            color: green;
        }

        #para1 {
            text-align: center;
            color: red;
        }

        .center {
            text-align: center;
            color: blue;
        }

        * {
            color: powderblue;
            font-size: 30px;
        }

        h1,
        h2,
        h3 {
```

```
            text-align: center;
            color: pink;
        }
    </style>
</head>

<body>
    <h1 class="center">Blue Center</h1>
    <p>paragraph</p>
    <p id="para1">Para1</p>
    <h2>Shalini</h2>
    <h3>Undergraduate Student</h3>
</body>

</html>
```

Blue Center

paragraph

Para1

Shalini

Undergraduate Student

4. Create a html page as shown below.

Author:

Title:

Price:

Submit Reset

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
```

```
    <title>Book Information</title>
  </head>
  <body>
    <form class="" action="index.html" method="post">
      <label for="author">Author:</label>
      <input type="text" id="author" name="author" value=""> <br> <br>
      <label for="title">Title:</label>
      <input type="text" id="title" name="title" value=""> <br> <br>
      <label for="price">Price:</label>
      <input type="number" id="price" name="price" value=""> <br> <br>
      <input type="submit" name="Submit" value="Submit">
      <input type="reset" name="Reset" value="Reset">
    </form>
  </body>
</html>
```

5. Design the below table using HTML

TIME TABLE

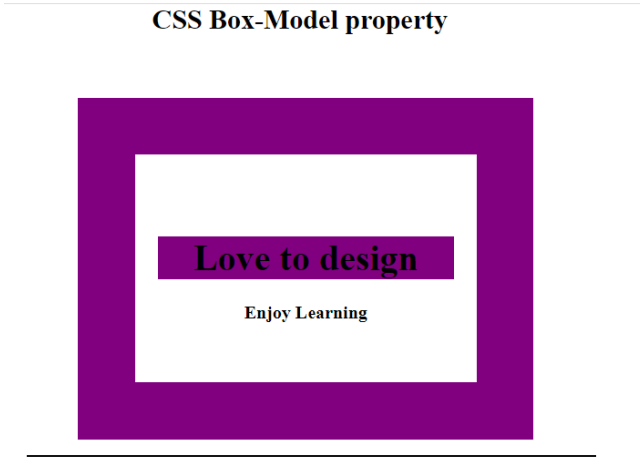
Day/Period	I 9:30-10:20	II 10:20-11:10	III 11:10-12:00	12:00-12:40	IV 12:40-1:30	V 1:30-2:20	VI 2:20-3:10	VII 3:10-4:00
Monday	Eng	Math	Chem	LUNCH	LAB			Phy
Tuesday	LAB				Eng	Chem	Math	SPORTS
Wednesday	Math	Phy	Eng		Chem	LIBRARY		
Thursday	Phy	Eng	Chem		LAB			Math
Friday	LAB				Math	Chem	Eng	Phy
Saturday	Eng	Chem	Math		SEMINAR			SPORTS

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Time table</title>
    <style media="screen">
      table{
        width: 100%;
      }
      tr{
        height: 50px;
      }
```

```
    table,tr,td,th{
      border-collapse: collapse;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1 style="text-align:center;">TIME TABLE</h1>
  <table border="5px">
    <tr>
      <th>Day/Period</th>
      <th>I <br> 9:30-10:20</th>
      <th>II <br> 10:20-11:10</th>
      <th>III <br> 11:10-12:00</th>
      <th> 12:00-12:40</th>
      <th>IV <br> 12:40-1:30</th>
      <th>V <br> 1:30-2:20</th>
      <th>VI <br> 2:20-3:10</th>
      <th>VII <br> 3:10-4:00</th>
    </tr>
    <tr>
      <th>Monday</th>
      <td>Eng</td>
      <td>Math</td>
      <td>Chem</td>
      <th rowspan="6">L <br>U <br>N <br>C <br>H</th>
      <td colspan="3">LAB</td>
      <td>Phy</td>
    </tr>
    <tr>
      <th>Tuesday</th>
      <td colspan="3">LAB</td>
      <td>Eng</td>
      <td>Chem</td>
      <td>Math</td>
      <td>SPORTS</td>
    </tr>
    <tr>
      <th>Wednesday</th>
      <td>Math</td>
```

```
      <td>Phy</td>
      <td>Eng</td>
      <td>Chem</td>
      <td colspan="3">LIBRARY</td>
    </tr>
    <tr>
      <th>Thursday</th>
      <td>Phy</td>
      <td>Eng</td>
      <td>Chem</td>
      <td colspan="3">LAB</td>
      <td>Math</td>
    </tr>
    <tr>
      <th>Friday</th>
      <td colspan="3">LAB</td>
      <td>Math</td>
      <td>Chem</td>
      <td>Eng</td>
      <td>Phy</td>
    </tr>
    <tr>
      <th>Saturday</th>
      <td>Eng</td>
      <td>Chem</td>
      <td>Math</td>
      <td colspan="3">SEMINAR</td>
      <td>SPORTS</td>
    </tr>
  </table>
 </body>
</html>
```

6 .Design the below CSS Box model using HTML and CSS properties




```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Box model</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="css/style.css" rel="stylesheet">
    <style>
        * {
            padding: 0px;
            margin: 0px;
            box-sizing: border-box;
        }

        .white {
            background-color: white;
            border-bottom: 2px solid black;
            width: 500px;
            height: 350px;
            margin: 20px;
        }

        .purple {
            background-color: purple;
            width: 400px;
            height: 300px;
            position: relative;
```

```
            top: 35px;
            left: 45px;
            padding: 50px;
            display: flex;
            justify-content: center;
            align-items: center;
        }

        .innerwhite {
            background-color: white;
            width: 350px;
            height: 200px;
            display: flex;
            justify-content: center;
            flex-direction: column;
            text-align: center;
        }

        .textpurple {
            background: purple;
            text-align: center;
            font-weight: 900;
            margin: 20px;
        }
    </style>

    <body>
        <h2 style="margin-left: 130px;">CSS Box-Model property</h2>
        <div class="white">
            <div class="purple">
                <div class="innerwhite">
                    <div class="textpurple">
                        <h1>Love to design</h1>
                    </div>
                    <p><b>Enjoy Learning</b></p>
                </div>
            </div>
        </div>
    </body>
</head>
```

```
<body>

</body>

</html>
```

7. Write a PHP script to display the contents of an employee table(empno, ename, esal) based on empno accepted through html document.

Php file

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Form 3</title>
  <style>
    table,th,td{
      border: 1px solid black;
    }
    th,td{
      padding: 10px;
    }
  </style>
</head>
<body>
  <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']);?>"
  method="POST">
  <h1>Search Employee form</h1>
  <label>Employee Number:</label>
  <input type="text" name="number" size="10" placeholder="Enter
  Number"><br><br>
  <input type="submit" name="save" value="Submit">
  <input type="reset" name="reset" value='Reset'>
  </form>
</body>
</html>
<?php
```

```
$host="localhost";
$username="root";
$password="";
$dbname="sample";
$conn = new
mysqli($host,$username,$password,$dbname);
if($conn->connect_error){
   die("connection failed" .$conn->connect_error);
}
else{
   if(isset($_POST["save"])){
      $number=$_POST["number"];
      $sql="SELECT * FROM employee WHERE number=$number";
      $result = $conn->query($sql);
```

```
        if($result->num_rows > 0){
            $row=$result->fetch_assoc();
            echo "<br><br><table><tr><th>Employee Number</th><td>".$row['number'].
"</td></tr><tr><th>Employee Name</th><td>".$row['name'].
"</td></tr><tr><th>Employee Salary</th><td>".$row['salary']. "</td></tr></table>";
        }else{
            echo "Error : Employee number is invalid";
        }
        $conn->close();
    }
  }
?>
```

Search Employee form

Employee Number: 7

Submit Reset

Employee Number	7
Employee Name	ram
Employee Salary	5000

8. Write a PHP script to accept employee number, name and salary from HTML document and display them. also insert the same into employee table.

Html file

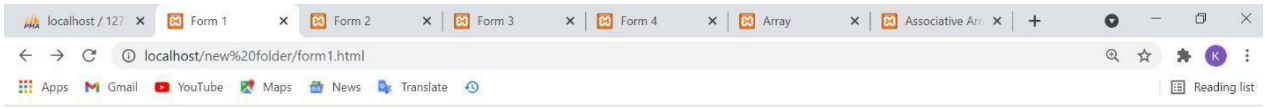
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Form 1</title>
</head>
```

```
<body>
  <form action="form1.php" method="POST">
  <h1>Insert Employee form</h1>
  <label>Number:</label>
  <input type="text" name="number" size="10" placeholder="Enter
  Number"><br><br>
  <label >Name:</label>
  <input type="text" name="name" placeholder="Enter Name"> <br><br>
  <label >Salary:</label>
  <input type="text" name="salary" size="10" placeholder="Enter Salary"><br><br>
  <input type="submit" name="save" value="Submit">
  <input type="reset" name="reset" value='Reset'>
  </form>
</body>
</html>
```

Php file

```
<!DOCTYPE html>
<html>
<head>
  <title>Form 1</title>
  <style>
    table,th,td{
      border: 1px solid black;
    }
    th,td{
      padding: 10px;
    }
  </style>
</head>
<body>
<?php
$host="localhost";
```

```
  $username="root";
  $password="";
  $dbname="sample";
  $conn = new
  mysqli($host,$username,$password,$dbname);
  if($conn->connect_error){
    die("connection failed" .$conn->connect_error);
  }
  else{
    if(isset($_POST["save"])){
      $number=$_POST["number"];
      $name=$_POST["name"];
      $salary=$_POST["salary"];
      $sql="INSERT INTO employee(number,name,salary) values
('$number','$name','$salary')";
      if($conn->query($sql)){
        echo "<table><tr><th>Employee Number</th><td>".$number.
"</td></tr><tr><th>Employee Name</th><td>".$name. "</td></tr><tr><th>Employee
Salary</th><td>".$salary. "</td></tr></table>";
      }else{
        echo "Error" .$sql. "<br>" .$conn->error;
      }
      $conn->close();
    }
  }
?>
</body>
</html>
```



Insert Employee form

Number: 7

Name: ram

Salary: 5000

Submit Reset

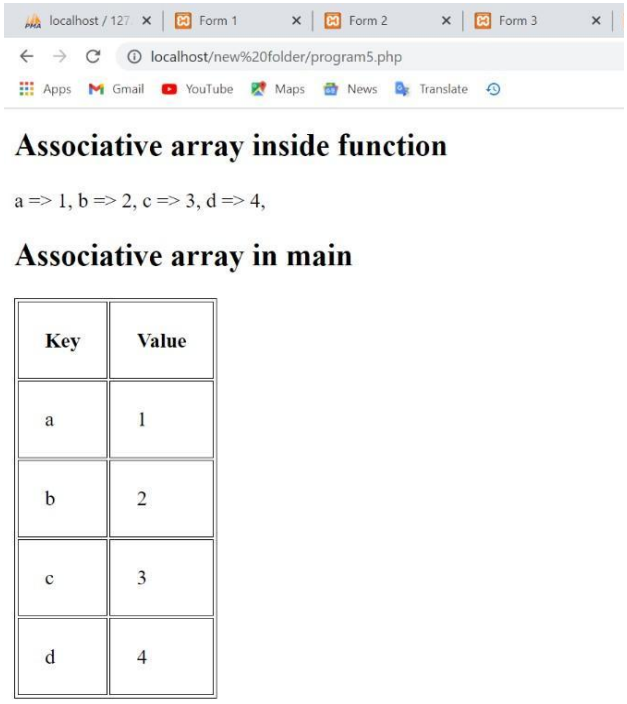
Employee Number	7
Employee Name	ram
Employee Salary	5000

9. Write a PHP program to create user defined function which returns Associative array. Display the contents in main program in table format.

Php file

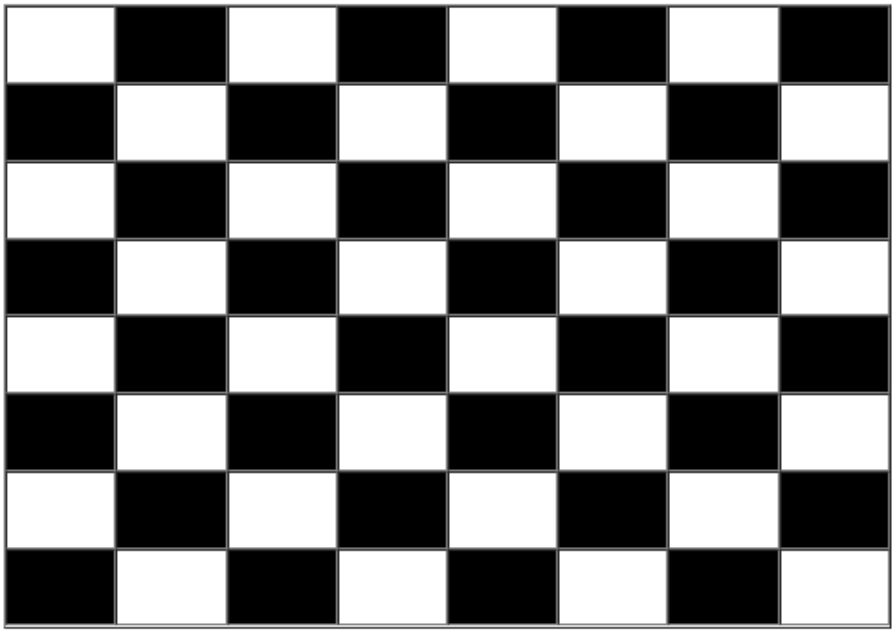
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Associative Array</title>
<style type="text/css">
table,th,td{
border: 1px solid black;
```

```
}
th,td{
   padding: 20px;
}
</style>
</head>
<body>
<?php

function associative(){
$arr=array("a"=>"1","b"=>"2","c"=>"3","d"=>"4");
echo "<h2>Associative array inside function<br></h2>";
foreach($arr as $x=>$y){
echo $x." => ".$y.", ";
}
echo "<br>"; return $arr;
}

$arr1=associative();
echo "<h2>Associative array in main<br></h2>";
echo "<table><tr><th>Key</th><th>Value</th></tr>";
foreach($arr1 as $x=>$y){
echo "<tr><td>".$x."</td><td>".$y."</td></tr>";}
echo "</table>"


?>
</body>
</html>
```



10. Write a program to create Chess board in PHP using for loop

```
<table width="400px" cellspacing="0px" cellpadding="0px" border="1px">
<?php
 for($row=1;$row<=8;$row++)
 {
  echo "<tr>";
 for($column=1;$column<=8;$column++)
{
$total=$row+$column;
if($total%2==0)
{
echo "<td height=35px width=30px bgcolor=#FFFFFF></td>";
}
else
{
echo "<td height=35px width=30px bgcolor=#000000></td>"; }
}
echo "</tr>"; }
?>
</table>
```



11. Demonstrate how to post data from HTML to PHP using php code.

Html file

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Form 2</title>
</head>
<body>
  <form action="form2.php" method="POST">
  <h1>Application form</h1>
  <label >Name:</label>
  <input type="text" name="name" placeholder="Enter name"> <br><br>
  <label>Age:</label>
  <input type="text" name="age" size="10" maxlength="3"><br><br>
  <label >Gender:</label><br><br>
  <input type="radio" name="gender" value="Male">Male<br>
  <input type="radio" name="gender" value="Female">Female<br><br>
  <input type="submit" name="save" value="Submit">
  <input type="reset" name="reset" value='Reset'>
  </form>
```

```
</body>
</html>
```

Php file

```
<!DOCTYPE html>
<html>
<head>
  <title>Form 2</title>
  <style>
    table,th,td{
      border: 1px solid black;
    }
    th,td{
      padding: 15px;
    }
  </style>
</head>
<body>
<?php
  if(isset($_POST["save"])){
```

```
    $name=$_POST["name"];
    $age=$_POST["age"];
    $gender=$_POST["gender"];
    echo "<table><tr><th>Employee name</th><td>".$name.
    "</td></tr>"; echo "<tr><th>Employee age</th><td>".$age.
    "</td></tr>";
    echo "<tr><th>Employee gender</th><td>".$gender. "</td></tr></table>";
  }
?>
</body>
</html>
```

Application form

Name: ram

Age: 21

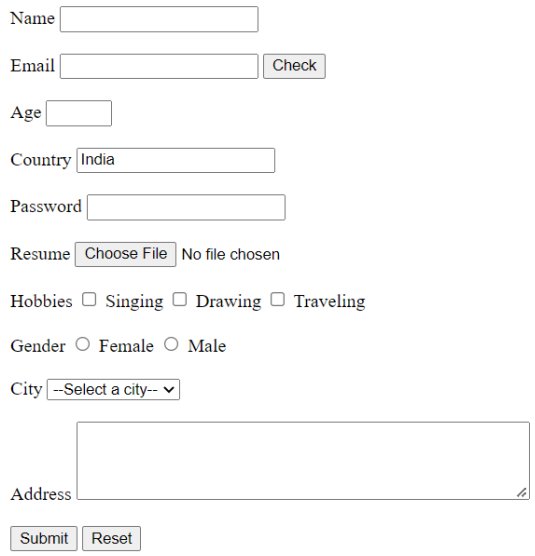
Gender:

(•) Male
( ) Female

Submit Reset

Employee name	ram
Employee age	21
Employee gender	Male

12. Create the below registration form using forms in HTML




```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Contact</title>
  </head>
  <body>
    <form class="" action="index.html" method="post">
      <label for="name">Name</label>
      <input type="text" id="name" name="name" value=""> <br> <br>
      <label for="email">Email</label>
      <input type="email" id="email" name="email" value="">
      <input type="button" name="check" value="Check"> <br> <br>
      <label for="age">Age</label>
      <input type="text" id="age" name="age" value="" maxlength="3" size="3"> <br>
<br>
      <label for="country">Country</label>
      <input type="text" id="country" name="country" value="India"> <br> <br>
      <label for="pwd">Password</label>
      <input type="password" id="pwd" name="pwd" value=""> <br> <br>
```

```
    <label for="resume">Resume</label>
    <input type="file" id="resume" name="resume" value=""> <br> <br>
    <label for="hobby">Hobbies</label>
    <input type="checkbox" id="hobby1" name="" value="">
    <label for="hobby1">Singing</label>
    <input type="checkbox" id="hobby2" name="" value="">
    <label for="hobby2">Drawing</label>
    <input type="checkbox" id="hobby3" name="" value="">
    <label for="hobby3">Traveling</label> <br><br>
    <label for="gender">Gender</label>
    <input type="radio" id="gender" name="gender" value="Female">
    <label for="gender">Female</label>
    <input type="radio" id="gender" name="gender" value="Male">
    <label for="gender">Male</label> <br><br>
    <label for="city">City</label>
      <select class="" id="city" name="city">
        <option value="--Select a city--" selected>--Select a city--</option>
        <option value="Delhi">Delhi</option>
        <option value="Mumbai">Mumbai</option>
        <option value="Hyderabad">Hyderabad</option>
        <option value="Madras">Madras</option>
        <option value="Bangalore">Bangalore</option>
      </select> <br><br>
    <label for="addr">Address</label>
    <textarea name="address" id="addr" rows="4" cols="50"></textarea> <br><br>
    <input type="submit" name="submit" value="Submit">
    <input type="reset" name="reset" value="Reset">
   </form>
  </body>
</html>
```