```
package Day4;

public class FirstProgram {

	public static void main(String[] args) {


		//  What is a class ??

		// Class is the place where we write our codes

		// Class is the blueprint of everything

		System.out.println("********************************");


		System.out.println(12345678);

		// Technically called as Integer
		// Integer can be marked as int

		// 123

		int a = 123 ;
				//123 Varaiable value
				// a Reference variable \
				// int is a data type

		 // 890\

		int b = 890 ;

		System.out.println(b);

		// LetteR  - A B C D E F G H

		//  It is said as char in java

		char e = 'A' ;

		// Words Eg ; Java  - Combination of characters are known as
string in java

		// String is a alphanumeric give in double quotes

		String c = "Java12234567654e&&&&&&&       ";

		// 970790299569

		int d = 970790298 ;

		long d1 = 9790299569L ;
```

```
            // True or False  // To say yes or no

            Boolean s = true ;


      }

}
```

---

```
package Day5;

public class MathematicalOperators {

      public static void main(String[] args) {

            // In your main method execution and compilation will happen

            // 1.1, 1.12

            // iN order to store decimal values

            float a = 1.1f ;

            //  + , - , * , /


            // 10 , 12

            int a1 = 10 ;

            int a2 = 12 ;

            int a3 = a1 + a2 ;

            System.out.println(a3);


            }

}
```

---

```
package Day5;

public class MethodsLearning {

      // OOPS - Object Oriented Programming structured language

      // Class - A place where we wil write our codes  // It is blueprint
of everything

      // Methods -- Actions or a Function that needs to be performed by a
porgramm is known as methods
```

```
	// Objects


	public void addit()  // Method declaration

	// Method Body
	{
		int x = 15 ;

		int y = 25 ;

		int z = x + y ;

		System.out.println(z);
	}


	}
---------------
package Day5;

public class RunnerCls {


	public void addit()  // Method declaration

	// Method Body
	{
		int x = 15 ;

		int y = 25 ;

		int z = x + y ;

		System.out.println(z);
	}

	// OOPS - Object Oriented Programming structured language

	// Class - A place where we wil write our codes  // It is blueprint
of methods and objects

	// Methods -- Actions or a Function that needs to be performed by a
porgramm is known as methods

	// Objects --  Is dependant on class name and main method
			// It is used to call the methods inside the class

	public static void main(String[] args) {



		// To access methods we need to create objects
```

```
            MethodsLearning ml1 = new MethodsLearning();
            ml1.addit();

            System.out.println("************************************");

            RunnerCls ra =new RunnerCls();
            ra.addit();
      }

}
```

---

```
package Day7;

public class ArithmeticOperators {

      public static void main(String[] args) {

            // Arithmetic Operators

            //  == (Comparing two data types)
            // <= Less than equal to
            // >= Greater than equal to
            // != Not equal to
            // < Lesser
            // > Greater

            int a = 10 ;
            int b = 20 ;

      Boolean resultforequal = a==b ;

      System.out.println(resultforequal);

  Boolean a1 =  a<=b ;

  System.out.println(a1);

  Boolean a2 =  a>=b ;

  System.out.println( a2);

  Boolean a3 = a<b ;
  System.out.println( a3);

  Boolean a4 = a>b ;
  System.out.println( a4);

  Boolean a5 = a!=b ;
  System.out.println(a5);
      }

}
```

---

```
package Day7;
```

```
import java.util.Scanner;

public class ScannerCls {

	public void addit()  // Method declaration

	// Method Body
	{
		int x = 15 ; // Hard Coding

		int y = 25 ;

		int z = x + y ;

		System.out.println(z);
	}

	public static void main(String[] args) {

		// Scanner is a class which is available in java

		// Interact with methods by giving inputs throught console

		Scanner sc = new Scanner(System.in);

		System.out.println("Enter the First Number");

		int x = sc.nextInt();

		System.out.println("The Entered number is" +x);

		System.out.println("Enter the Second Number");

		int y = sc.nextInt();

		System.out.println("The Entered number is" +y);

		int z = x + y ;

		System.out.println("The addition value of this is" +z);
	}

}
---------------
package Day7;

public class StringsTopics {

	public static void main(String[] args) {

		// Concation - Adding two diferent data types , Strings

	 // +
```

```
	// +
		//  Navin Kumar Sampath

		String fname = "Navin" ;
		String mname = "Kumar" ;
		String lname = "Sampath" ;

		//  My full name is

		System.out.println("My full name is " + fname+ " " + mname+ "
" + lname);



	}

}
------------------------------
package Day8;

public class MoreMethods {

	//  We are going to see more types of methods

	// Design ur assignments accordingly with the different types of
methods

	// Return Type - WHat data type its going to return

	//  Void - It will not retrun anything it is not going to return
anything because it is not a data type

	public int add()
	{
		int a = 10 ;
		int b = 15 ;
		int c = a +b ;
		return c ;
	}

	public int sub()
	{
		int d = 20 ;
		int e = 10 ;
		int f = d-e ;
		return e ;
	}

	public String names()
	{
		String fname = "Navin" ;
		String mname = "Kumar" ;
		String lname = "Sampath" ;
```

```
            return fname+ " " + mname + "   " + lname ;
      }



}
----------
package Day8;

public class MoreMethods1 {

      public static void main(String[] args) {
            MoreMethods mm =  new MoreMethods();

            int a = mm.add() ;
            System.out.println(a);

            int b = mm.sub();
            System.out.println(b);

            String n = mm.names();
            System.out.println(n);

      }

}
--------
package Day8;

public class MoreonStrings {

      public static void main(String[] args) {

            // String is a non primitive data types
            // iT is said to be as class  // It has methods in it
            //  Strings are not explicitey created


            String s = "This is my first java class    " ;
                  //      12345678901234567890123456789012345678901

                  int n = s.length();
                  System.out.println(n);

            String s1 = s.trim();
      int n1 =    s1.length();

      System.out.println(n1);
      }

}
---------------------------------
package Day9;
```

```
public class LengthVsIndexing {

	public static void main(String[] args) {

		// What is length ??
		// Total number of charcter in a string
		// It will start from 1

		//What is indexing ??
		// Length - 1
		//  It will start from 0


		String s = "This is the java class       " ;
		      //          12345678901234567890123456789
		System.out.println(s.length());

		String s3 = "This is the java class       " ;
		 //          012345678901234567890123456789

		String s1 = "Navin" ;
		String s9 = "navin" ;

	String newstring  =s1.concat(s9) ;

	System.out.println(newstring);

		Boolean y = s1.equals(s9) ;
		Boolean n = s1.equalsIgnoreCase(s9);



		System.out.println(n);

		char o = s3.charAt(21) ;
		System.out.println(o);

		// 03-07-2023

		String date = "03-07-2023" ;

		System.out.println(date);

		// 03/07/2023

		String update = date.replace('-', '/');

		System.out.println(update);


//		String s3 = "This is the java class       " ;
		 //          012345678901234567890123456789
```

```
            //  Subsbtring
            String whichclass = s3.substring(12, 16) ;

            System.out.println(whichclass);

            String anothersubstring = s3.substring(12) ;
            System.out.println(anothersubstring);



            Boolean conta = s3.contains("java") ;
            System.out.println(conta);


      }

}
---------
package Day9;

public class StringsLearing {

      public static void main(String[] args) {

            // *** Interview Questions****


            // Why String is Immutable In Nature :

            // Strings once created its created we cant replace or mute
it

            // Strings uses concept called string pool which makes it as
immutable

            //  Whenever you are creating a varaible values which is
already created mapping for your varaible in string pool is changed

            //  New memory will not be allocated for the existing memory
it self









      }

}
-------------------------------------------------
package Day10;

public class Arrays {
```

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


            String s1 = "Chaitra" ;
            String s2 = "Jyothi" ;
            String s3 = "Mrunali" ;
            String s4 = "Sindhuja" ;
            String s5 = "SreeVidhya";
            String s6 = "Vidhya" ;
            String s7 = "Prsanth" ;
            String s8 = "Shenba" ;
            String s9 = "Zenith";
            String s10 ="NavinSampath" ;

            int s1id = 101 ;
            int s2id = 102 ;
            int s3id = 103 ;
            int s4id = 104 ;
            int s5id = 105 ;
            int s6id = 106 ;
            int s7id = 107 ;
            int s8id = 108 ;
            int s9id = 109;
            int s10id = 110 ;


            /// My id is ** and My name is **

            System.out.println("My id is"+" " +s1id+" " +s1);
            System.out.println(s2);
            System.out.println(s3);
            System.out.println(s4);
            System.out.println(s5);
            System.out.println(s6);
            System.out.println(s7);
            System.out.println(s8);
            System.out.println(s9);
            System.out.println(s10);


            System.out.println("First way of adding Strings names to
array ");
            String [] studentnames = new String[20] ;
            studentnames[0] = "Chaitra" ;
            studentnames[1] = "Jyothi" ;
            studentnames[2] = "Mrunali" ;
            studentnames[3] = "Sindhuja" ;
            studentnames[4] = "SreeVidhya";

            System.out.println("Second way of adding Strings names to
array ");
```

```
            String [] studentnames1 =
{"Chaitra","Jyothi","Mrunali","Sindhuja","SreeVidhya"
,"Vidhya","Prsanth","Shenba", "Zenith" ,"NavinSampath"};

            System.out.println(studentnames.length);
            System.out.println(studentnames1.length);

            System.out.println(studentnames[4]);

            System.out.println(studentnames[15]);

            System.out.println(studentnames[0]);


            System.out.println(studentnames1[4]);
            System.out.println(studentnames1[9]);
            // Arrays will work on basis of index
            // Once size is declared we cant increase or decrease the
size

      }

}
----------
package Day10;

public class ControlStatement {

      public static void main(String[] args) {

            /*
             * if() {
             *
             * } else if() {
             *
             * } else if() {
             *
             * } else if() {
             *
             * } else {
             *
             * }
             */

            int Chem = 189 ;
            int phy  = 200 ;
            int bio = 200 ;
            int comp = 200 ;
            int maths = 199 ;

            int mbbs = bio + phy + Chem ;
            int be = Chem + phy  +maths ;

            if(mbbs == 600)
```

```
            {
                  System.out.println("Selcted for MBBS");
            }

            else if(be == 600)
            {
                  System.out.println("Selected for top level Eng Clg");
            }
            else
            {
                  System.out.println("Selected for Normal Enginering
clg");
            }

      }

}
-------------
package Day10;

public class Loops {

      public static void main(String[] args) {

            System.out.println("************************");
            System.out.println(1);
            System.out.println(2);
            System.out.println(3);
            System.out.println(4);
            System.out.println(5);
            System.out.println(6);
            System.out.println(7);
            System.out.println(8);
            System.out.println(9);
            System.out.println(10);
            System.out.println("************************");


            // Loops - For Loop - Iteration

            // Declaring // Intilization ---- int a = 1

            // Condition a<=10

            // Increment a++ , a--

            // Print 799 below 1099

            // Print 1000 to 1 in descending order

            // Create int array and save ur id

            // a = 0 ;
```

```
            // a <= studentnames.length

            // a ++

            // studentnames[a]

            for(int a = 1 ; a<=10 ; a++)
            {
                  System.out.println(a);
            }

      }

}
---------
package Day10;

public class WhileLoop {

      public static void main(String[] args) {

            // Initilization
            // Condition
            // Increment

            int a = 1 ;
            int b = 100 ;

            // a<=100

             // a ++ ;

            while(a<=100)
            {
                  System.out.println(a);
                  a++;
            }


      }

}
-----------------------------------
package Day14;

public class LogicalOperators {

      // Logical Operators


      // && AND

      // || OR
```

```
	// !=


	// || OR


	//  T T - T
	// T - F - T
	// F- T - T
	// F - F - F


	// &&

	// T T = T
	// F T = F
	// T F = F
	// F F = F


}
------
package Day14;

public class LopicalOpWorking {

	public static void main(String[] args) {


		int a = 10 ;
		int b = 20 ;
		int c = 30 ;


		// * Condtion 1 - To verify a < b and b > c [ Any 1 is true
cond 1 is satfied ]
				// Contion 1 Sataisfied

		// Condtion 2 - To verify a < b and c > b [ Both satttement
should be true]
				// Condtion 2 satisfied

		//    T && F  = F
		if(a<b && b>c  ) // F
		{
			System.out.println("Condtion 1 satisfied");
		}
			//    T || F = T
		else if(a<b  || c<b)
		{
			System.out.println("Condtion 2 satisfied");
		}
		else
		{
```

```
                System.out.println("Nothing is true");
            }



      }

}
-------
package Day14;

public class CollectionLearning {


      // Drawbacks of Array

      // Fixed Size, Similar data types

      // Collection is a Interface -

      //  Which will overcome the drawbacks of arrays conecpt

      //  Collection size is not fixed
      // Collection size can be increased gradually
      // Based on the user defined it can accept all data type in one
collection
      // If you want to restrict it can accept only strng only int too

      // Collection is sp;iited up into three types of interfaces

      //  List [I]
      // SET [I]
      // MAP [I]


      //  List is an Interface which is implemeted in two following
classes

      // Array List  [Classes]

      // Linked List [CLasses]



}
------
package Day14;

import java.util.ArrayList;

public class ArrayListLearning {

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

```
String[] studnames = {"Navin" , "Jones" , "Sampath"} ;


ArrayList<String> studnameb6 = new ArrayList<String>();
studnameb6.add("Navin");
studnameb6.add("Kumar");
studnameb6.add("Sampath");
studnameb6.add("Nivaan");
studnameb6.add("Arjun");
studnameb6.add("Shyam");

// Array List will work on the basis of indexing

// Size
System.out.println(studnameb6.size());

// To retrive using index
System.out.println(studnameb6.get(5));

// To remove some objects using index
System.out.println(studnameb6.remove(5));

// Size
System.out.println(studnameb6.size());

//// To remove some objects using object name

System.out.println(studnameb6.remove("Arjun"));

// Size
System.out.println(studnameb6.size());

// Contains

System.out.println(studnameb6.contains("NIvaan"));


// Contains

System.out.println(studnameb6.contains("Nivaan"));


for(String a : studnameb6)
{
      System.out.println(a);
}

ArrayList<Object> alldattype = new ArrayList<Object>();

alldattype.add(123445) ;
alldattype.add('c');
alldattype.add("Stringadded");
alldattype.add(5.9);
```

```
            for(Object o : alldattype)
            {
                  System.out.println(o);
            }


      }

}
```

---

```
package Day15;

import java.util.LinkedList;

public class LinkesList {

      public static void main(String[] args) {

            // List is Interface which Implements to the following class

            //  Array List

             // Linked List

            //  Linked List - To increase the performance by using data
structures conecpts

            //  The retrieving speed of Linked list using index is
sppeder than in array list which is in mili seconds

            // Linked list also works on basis of indexing


            LinkedList<String> names = new LinkedList<String>();
            names.add("Chaitra");
            names.add("Jyoti");
            names.add("Murnali");
            names.add("Prsaanth");
            names.add("Sarath");
            names.add("Sham");
            names.add("Sinhu");
            names.add("Vidhya");
            names.add("Zenith");
            names.add("Arjun") ;
            names.add("Jyoti") ;



            // Size

            System.out.println(names.size());

            // Remove using object
```

```
            System.out.println(names.remove("Arjun"));

            // Size

            System.out.println(names.size());


            for(String a : names) {
                  System.out.println(a);
            }

            LinkedList<Object> alldatyp = new LinkedList<Object>();

            alldatyp.add(1);
            alldatyp.add("Strings added");
            alldatyp.add('c');
            alldatyp.add(7.9);
            alldatyp.add(true);

            for(Object o : alldatyp)
            {
                  System.out.println(o);
            }




      }

}
------
package Day15;

import java.util.HashSet;

public class SetLearning {

      // Set - Interface from collection interface

      // Which will not allow you duplicate values

      // Duplicates values are ignored at the compilation or ignored

      // Set interface is implemeted in the following classes

      // Hashset [ Class ]

      // Linked Hash Set [ Class]

      public static void main(String[] args) {

            HashSet<String> languages = new HashSet<String>();
            languages.add("Java");
            languages.add("English");
```

```
            languages.add("Tamil");
            languages.add("Hindi");
            languages.add("Java");
            languages.add("Python");
            languages.add("C++");


            //  Hashset

            //  It will allow to save duplicates but it wil be ignored at
the time of compilation
            //  Insertion order is not maintained
            //  THERE WLL BE NO GET METHOD IT DOES NOT MAINTAIN INSERTION
ORDER

            for(String e : languages) {
                  System.out.println(e);
            }

            System.out.println(languages.size());

            System.out.println(languages.contains("Java"));


            System.out.println(languages.remove("Python"));

            System.out.println(languages.size());


      }

}
-----
package Day15;

import java.util.LinkedHashSet;

public class LinkedHassetlearning {

      public static void main(String[] args) {

            //  Linkedhasset

            // It will not allow duplicates
            //  It will maintain insertion order
            //  There will be no get (index) method


            LinkedHashSet<String> names = new LinkedHashSet<String>();

            names.add("Java");
            names.add("English");
            names.add("Navin");
            names.add("Java");
```

```
            names.add("Nivaa");
            names.add("Jones");
            names.add("Sampath");
            names.add("Maths");

            System.out.println(names.size());

            System.out.println(names.remove("English"));

            System.out.println(names.contains("Jones"));

            for(String s : names)
            {
                  System.out.println(s);
            }




      }

}
----------------------------------------
package Day16;

import java.lang.reflect.Array;
import java.util.ArrayList;

public class MapInterfaceLearing {

      public static void main(String[] args) {

            // String -- //Integer

            // Social  - 80

            // Science  -90

            // Maths - 100

            // Hindi - 90

            // English -95

            // Tamil -90


            ArrayList<String> subj = new ArrayList<String>();
            subj.add("Social");
            subj.add("Science");
            subj.add("Maths");
            subj.add("Hindi");
            subj.add("English");
            subj.add("Tamil");
```

```
            ArrayList<Integer> marks = new ArrayList<Integer>();
            marks.add(80);
            marks.add(90);
            marks.add(100);
            marks.add(90);
            marks.add(95);
            marks.add(90);

            // I need a output Saying that Marks scored in XXXXX is XX

            for(int i = 0 ; i<subj.size(); i++)
            {
                  System.out.println("The Marks Scored in " +subj.get(i) +
" is " +marks.get(i));
            }

      }

}
--------
package Day16;

import java.util.HashMap;

public class LinkedHashMap {

      public static void main(String[] args) {


            // Map is a interface which will store in key and value
format

            //  Map is Interface

            //  HashMap [Class]

            // Linked HashMap [Class]\

            // String -- //Integer

                        // Social  - 80

                        // Science  -90

                        // Maths - 100

                        // Hindi - 90

                        // English -95

                        // Tamil -90
```

```
            //HashMap - It will not store in ordered list
            // Keys will not allow u to save duplicates
            // Values will allow u to save duplicates

            HashMap<String, Integer> markdetails = new HashMap<String,
Integer>();
            markdetails.put("Social", 80);
            markdetails.put("Science", 90);
            markdetails.put("Maths", 100);
            markdetails.put("HIndi", 90);
            markdetails.put("English", 95);
            markdetails.put("Tamil", 90);

            //  To print the size
            System.out.println(markdetails.size());

            // We can get the values using get(keys) method
            System.out.println(markdetails.get("English"));

            // To replace
            System.out.println(markdetails.replace("Tamil", 90, 85));

            for(String e : markdetails.keySet())
            {
                  System.out.println("The Marks Scored in "+ e + " is "+
markdetails.get(e));
            }




      }

}
----
package Day16;

import java.util.LinkedHashMap;

public class MapPart2 {

      public static void main(String[] args) {

            // Linked HashMap - It uses data structur concpets

                        // It save in ordered list
                        // Keys will not allow u to save duplicates
                        // Values will allow u to save duplicates

            LinkedHashMap<String, Integer> empdetails = new
LinkedHashMap<String, Integer>();
            empdetails.put("Navin", 101);
            empdetails.put("Nivaan", 102);
```

```
            empdetails.put("Sindhu", 103);
            empdetails.put("Mrunali", 104);
            empdetails.put("Vishya", 105);

            System.out.println(empdetails.size());

            empdetails.replace("Nivaan", 102, 100);

            for(String c : empdetails.keySet())
            {
                  System.out.println("The Emp Name is " +c + " and the id
no is " +empdetails.get(c));
            }

      }

}
```

```
package Day12;

public class AccessModifiers {

	//  Public - Publically it is avaiable everywhere

	// Access modifiers Can be used in data type / methods

	// Default  - Without any access modifers that methods are said be as feault access modiers methods

	//  This can be used inside the package, we cant use this outside the package


	// Protected - Protected and default access mofiders are same

	// it can be used inside the package cannot be used outside the package

	// Final - we can just view where ever we want we cant overide or replace data types or methods


	//  Private - Is a separate topic which we will be seeing encapsulation

	void add ()
	{
		int a = 10 ;
		int b = 20 ;
		int c = a +b ;
		System.out.println(c);
	}

	public void sum()
	{
		int a = 10 ;
		int b = 20 ;
		int c = a - b ;
		System.out.println(c);
	}

	protected void div()
	{
		int a = 10 ;
		int b = 20 ;
		int c = a / b ;
		System.out.println(c);
	}

	int a = 20 ;

	final int  b = 20 ;



}
---------
package Day12;

public class PolyMorphismPart2 {
```

```
	//Method Overriding - Dyanamic polymporshim / Run time polymoprhism

	public void add()
	{
		int a = 10 ;
		int b = 20 ;

		int c = a +b ;

		System.out.println(c);
	}

	// The addition of two numbers is c

	//  By extending the class to the sub class and we can overidid the
implementation

	// By using overriding concept your code will give preference to the
overidedn methods

	public void sub()
	{
		int a = 10 ;
		int b = 20 ;

		int c = a -b ;

		System.out.println(c);
	}
}
package Day12;

public class MethodOverdiding extends PolyMorphismPart2{

	public void add()
	{
		int a = 10 ;
		int b = 20 ;

		int c = a +b ;

		System.out.println("The addiition of two numbers is " +c);
	}


}
-------
package Day12;

public class RunnerClass {

	public static void main(String[] args) {
		MethodOverdiding mo = new MethodOverdiding();
		mo.add();
		mo.sub();

		AccessModifiers am = new AccessModifiers();
```

```
            am.add();
            am.sum();
            am.div();

            System.out.println(am.a);
            am.a = 40 ;
            System.out.println("After overiding");
            System.out.println(am.a);
            System.out.println(am.b);

             // am.b = 50 ;
      }

}
----------------------------
package Day12;

public class Encapsualtion {

      //  Binding datas with the codes are said to be encapsualtion

      //  Encpasualtion can be achieved by setting your datatypes access modifers
as Private

      //  If we set the  datatypes access modifers as Private we are gotta use
Getter and Setter method to access those datas



}
package Day12;

public class RBI {

      private long accountno ;

      private long phno ;

      private int loanno;

      private String address ;

      public long getAccountno() {
            return accountno;
      }

      public void setAccountno(long accountno) {
            this.accountno = accountno;
      }

      public long getPhno() {
            return phno;
      }

      public void setPhno(long phno) {
            this.phno = phno;
      }

      public int getLoanno() {
```

```
		return loanno;
	}

	public void setLoanno(int loanno) {
		this.loanno = loanno;
	}

	public String getAddress() {
		return address;
	}

	public void setAddress(String address) {
		this.address = address;
	}



}
package Day12;

public class CANARA {

	public void setcanaradetails()
	{
		RBI r = new RBI();
		r.setAccountno(1010019292384L);
		r.setAddress("Chennai TamilNadu");
		r.setLoanno(190986);
		r.setPhno(9790299569L);


		System.out.println(r.getAccountno());
		System.out.println(r.getAddress());
		System.out.println(r.getLoanno());
		System.out.println(r.getPhno());

	}

}
package Day12;

public class HDFC  {


	public void setdetails()
	{
		RBI r = new RBI();
		r.setAccountno(1010019292384L);
		r.setAddress("Vellore TamilNadu");
		r.setLoanno(190986);
		r.setPhno(9790299569L);


		System.out.println(r.getAccountno());
		System.out.println(r.getAddress());
		System.out.println(r.getLoanno());
		System.out.println(r.getPhno());

	}
```

```
}
package Day12;

public class SBI {


	public void setsbidetails()
	{
		RBI r = new RBI();
		r.setAccountno(1010019292384L);
		r.setAddress("Chennai TamilNadu");
		r.setLoanno(190986);
		r.setPhno(9790299569L);


		System.out.println(r.getAccountno());
		System.out.println(r.getAddress());
		System.out.println(r.getLoanno());
		System.out.println(r.getPhno());

	}

}
package Day12;

public class rbirunnerclass {

	public static void main(String[] args) {


		HDFC h = new HDFC();

		h.setdetails();

	System.out.println("**************************************************");
		SBI s = new SBI();

		s.setsbidetails();

	System.out.println("**************************************************");

		CANARA c = new CANARA();
		c.setcanaradetails();

	}

}
-------------------------------
```

```
package Day13Interface;

public interface InterfaceLearning {

	// INterface - Set of rules that needs to be followed 100 %

	// Using interface we can achieve multiple inheritance

	// 100 % Abstarction is achieved here in interface

	//  You are not alowed to use  non abstarct method

	// We cant create objects for interface

	// What ever you are creating in interface is abstarct in nature

	//  No abstarct keyword is necessary

	//  implements keyword which is used to inherit properties from
parent to achieve multiple inheritance



}
package Day13Interface;

public interface OfficeRules {


	// What ever you are creating in interface is abstarct in nature

	public void wearid() ;
	public void dresscode();
	public void logintimings();
	public void norashdriving();
	public void takecareofyourassets();

}
package Day13Interface;

public interface CafteriaRules {

	public void maintainsilence();
	public void maintainque() ;
	public void throwtrashinbins();

}
package Day13Interface;

public class SignonMethods implements OfficeRules , CafteriaRules {

	public void maintainsilence() {
```

```
            System.out.println("Signing on Maintain Silence in
Cafteria");

      }

      public void maintainque() {
            System.out.println("Signing on Maintain que in Cafteria");

      }

      public void throwtrashinbins() {
            System.out.println("Signing on to Throw tash in respectve
bins in Cafteria");

      }

      public void wearid() {
            System.out.println("Signing on to Wear id on office all the
time ");

      }

      public void dresscode() {
            System.out.println("Signing on to Follow dress code all the
days ");

      }

      public void logintimings() {
            System.out.println("Signing on to follow login timings in
office all the time ");

      }

      public void norashdriving() {
            System.out.println("Signing on to say No Rash driving in
office  ");

      }

      public void takecareofyourassets() {
            System.out.println("I wil safe guard th eoffice assets ");

      }

}
package Day13Interface;

public class Runner {

      public static void main(String[] args) {
            SignonMethods sm = new SignonMethods();
            sm.dresscode();
            sm.norashdriving();
```

```
            sm.logintimings();
            sm.wearid();
            sm.maintainque();
            sm.maintainsilence();
      }

}
```

---

Abstract

```
package Day13;

public class AbstractionLearning {


      // Multiple inheritance is not allowed in java

      // Abstarction - Starting approach to fix multiple inheritance was
abstraction

      // Set of rules which needs to be followed

      // 100 % Multiple inhertance is not achieved in abstarction concept



}
package Day13;

public abstract class AbstractLearn {


      // Multiple inheritance is not allowed in java

      // Abstarction - Starting approach to fix multiple inheritance was
abstraction

      // Set of rules which needs to be followed

      // 100 % Multiple inhertance is not achieved in abstarction concept

      // It allows non abtsarct methods as well

      // For creating abstract class or methods abstract keyword is
important is needed

      // for abstarct methods it will not allow you to write
implementation

      // We cant create object for abstarct class and interface

      //  Non abstract methods

      public void display ()
      {
```

```
			System.out.println("Display Numbers 1234567810");
		}

		public abstract void add();
		public abstract void sum();
		public abstract void div();



}
package Day13;

public class MethodsforAbstraction extends AbstractLearn {

		@Override
		public void add() {

			int a = 10 ;
			int b = 20 ;

			System.out.println(a+b);

		}

		@Override
		public void sum() {
			int a = 10 ;
			int b = 20 ;

			System.out.println(a-b);


		}

		@Override
		public void div() {
			int a = 10 ;
			int b = 20 ;

			System.out.println(a/b);

		}

}
package Day13;

public class Runner {

		public static void main(String[] args) {

			MethodsforAbstraction mm = new MethodsforAbstraction();
			mm.add();
			mm.sum();
			mm.display();
```

```
            mm.div();
        }
    }
```

```
package Day17;

public class ExceptionsHandleLearnig {

	// Excpetion : Error which is throw by ur java

	// Array Index out of bound execption
	// String index out of bound execption
	//  Input mismatch exepction

	// Execption Handling

	// We are going to try some action

	// - If that action is getting failed

	// Catch the excpetion and display continue wuth the execution



}
package Day17;

import java.util.Scanner;

public class ExecptionHandlingMethods {

	public void add()
	{

		try {
			Scanner s = new Scanner(System.in);
			System.out.println("ENter Number a ");
			int a = s.nextInt();
			System.out.println("ENter Number b ");
			int b = s.nextInt();
			int c = a +b ;
			System.out.println(c);
		}
		catch(Exception e)
		{
			System.out.println(e);
		}
		finally
		{
			System.out.println("I will be executed in alll case");
		}

	}

	public void sub()
	{
		try {
```

```
                Scanner s = new Scanner(System.in);
                System.out.println("ENter Number a ");
                int a = s.nextInt();
                System.out.println("ENter Number b ");
                int b = s.nextInt();
                int c = a - b ;
                System.out.println(c);
          }
          catch (Exception e) {
                System.out.println(e);
          }
          finally
          {
                System.out.println("I will be executed in alll case");
          }

    }

    public static void main(String[] args) {
          ExecptionHandlingMethods em = new ExecptionHandlingMethods();
          em.add();
          em.sub();
    }

}
--------
package Day17;

public class InsufficientBalanceException extends Exception{


    public InsufficientBalanceException()
    {

          System.out.println("Enter Less Amount Because the Balance is
Insuffienct ");
    }

}
package Day17;

import java.util.Scanner;

public class Atmwhithdrwal {

    // Throw is said to be as user defined execption
    // Throws is said to be a java execption - Which will not allow you
to execute the program

    int bal = 1000 ;

    public void debit()
    {
          System.out.println("Enter the amount to be Withdrawl");
```

```
            Scanner s = new Scanner(System.in);
            int with = s.nextInt();

            try
            {
                  if(with > bal)
                  {
                        throw new InsufficientBalanceException();
                  }
            }
            catch(InsufficientBalanceException e)
            {
                  System.out.println(e);
            }
      }

}
package Day17;

public class RunnerClass {

      public static void main(String[] args) {
            Atmwhithdrwal a = new Atmwhithdrwal();
            a.debit();

      }

}
```

```
package Day18;

public class FFF {

	// Final : Values cannot be overriden,

	// Finally : Will be coming under exception handling hierarchy, At any cost try or catch is exeucted finally will beexecuted

	// Finalize : Finalize is used in garbage collector which removes memory for unused varibles

	public static void main(String[] args) {

		int num ;

		System.gc();



	}

}
---------------
package Day18;

import java.util.ArrayList;
import java.util.Collections;

public class CollectionVSCollections {

	// Collection : Is an interface

	// Collections : It is an Class which comes from JAVA which has inbuild methods

	// Assignment :

	// Arraylist which is an Integer, dont add any numbers
	// Using for loop you have to add numbers from 0 to 15 in the created array list
	// Print the array list in descending order using for loop , Collections

	public static void main(String[] args) {

		ArrayList<Integer> marks = new ArrayList<Integer>();

		marks.add(99);
		marks.add(80);
		marks.add(100);
		marks.add(89);
		marks.add(99);
		marks.add(67);
```

```
            for(Integer e :marks )
            {
                  System.out.println("The Marks scored is " +e);
            }


      System.out.println("**********************************************
**");

            // Traditional Loop

            // Intilization int i = marks.siez()-1
            // i>=0 ;
            // i--

            for( int i = marks.size()-1 ; i >= 0 ; i--)
            {
                  System.out.println(marks.get(i));
            }


      System.out.println("**********************************************
**");

            for(Integer e :marks )
            {
                  System.out.println("The Marks scored is " +e);
            }


      System.out.println("**********************************************
**");

            // Collections is a class which is having methods

            Collections.reverse(marks);

            for(Integer e :marks )
            {
                  System.out.println("The Marks scored is " +e);
            }


      }

}
------------------
package Day18;

public enum Daysinaweek {

      Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday
```

```
}
-----
package Day18;

public class ThrowVsThrows {

	// Throw : Throw is a user defined Exceptions , We can continue the
executions

	// Throws : Throws is a exection throwned by ur JAVA, We cant
continue to execution

	public static void main(String[] args) {


		// Accesing Enum

		for(Daysinaweek d : Daysinaweek.values())
		{
			System.out.println("The days in a weeks are " + d);
		}

	}

}
```

```
package Day19;

import java.io.File;

public class CreatingFiles {

	public static void main(String[] args) {

		File f = new File("C:\\Users\\Dell\\eclipse-
workspace\\JavaTrainingb6\\src\\test\\resources\\demo.txt");

		try {

			if(f.createNewFile())
			{
				System.out.println("File has been created");
			}
			else
			{
				System.out.println("File is alread created ");
			}


		} catch (Exception e) {System.out.println(e);// TODO: handle
exception
		}


	}

}
----------------
package Day19;

import java.io.File;
import java.io.FileWriter;

public class WritingFiles {

	public static void main(String[] args) {

	File f = new File("C:\\Users\\Dell\\eclipse-
workspace\\JavaTrainingb6\\src\\test\\resources\\demo.txt");

	FileWriter fw ;

	try {
		fw = new FileWriter(f);
		fw.write("This is Last session of java Hope u guys enjoyed
it");
		fw.close();

	} catch (Exception e) {
		System.out.println(e);
```

```
            // TODO: handle exception
      }


      }

}
----------------
package Day19;

import java.io.File;
import java.io.FileReader;
import java.util.Scanner;

public class ReadingFiles {

      public static void main(String[] args) {

            File f = new File("C:\\Users\\Dell\\eclipse-
workspace\\JavaTrainingb6\\src\\test\\resources\\demo.txt");
            FileReader fr ;

            try {
                  fr = new FileReader(f);
                  Scanner s = new Scanner(f);

                  if(s.hasNextLine())
                  {
                        System.out.println(s.nextLine());
                  }
                  else
                  {
                        System.out.println("File is not presented");
                  }

            } catch (Exception e) {
                  System.out.println(e);
            }

      }

}
```