**JAVA PROGRAMS**

1. **DATA TYPES**

**package** com.java;

**public** **class** DataTypes {

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

**int** a = 123;

**char** c = 'g';

**byte** b = (**byte**)32;

**short** s = (**short**)4356;

**long** l = 52143624l;

**float** f = 53.6f;

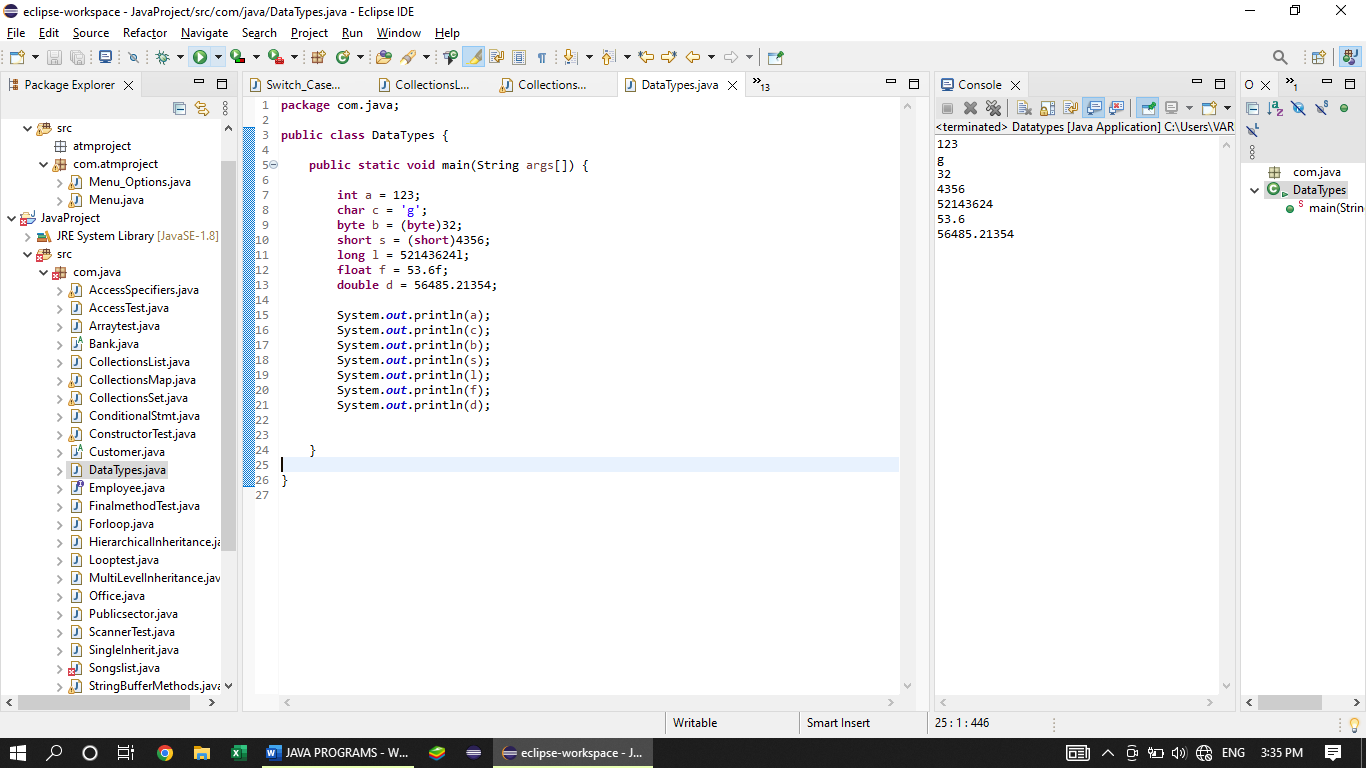
**double** d = 56485.21354;

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

}

}

**OUTPUT:**



1. **ENCAPSULATION**

**package** com.java;

**public** **class** AccessSpecifiers {

**public** **void** BankSchemes() {

**long** minbls = 1000l;

String monthlyint = "5%";

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

}

**private** **void** AccountDetails() {

**long** managersal = 50000l;

**long** housekeepingsal = 10000l;

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

}

**protected** **void** PinDetails() {

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

}

**void** Banner() {

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

}

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

AccessSpecifiers obj = **new** AccessSpecifiers();

obj.BankSchemes();

obj.AccountDetails();

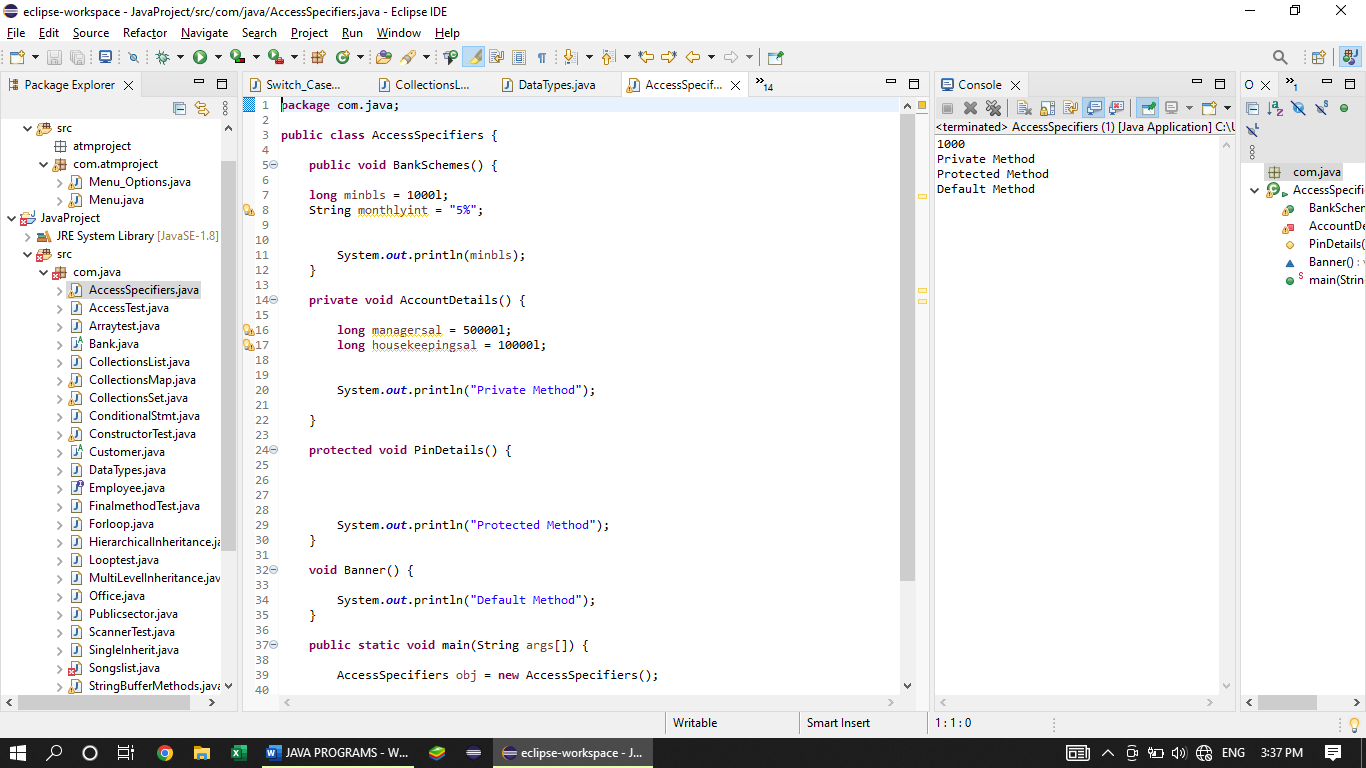
obj.PinDetails();

obj.Banner();

}

}

**OUTPUT:**



1. **INHERITANCE**
2. **SINGLE INHERITANCE**

**package** com.java;

**public** **class** SingleInherit **extends** Customer{

@Override

**public** **void** schemes() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Bank");

}

@Override

**public** **void** jewelloan() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Bank");

}

@Override

**public** **void** cashcounter() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Bank");

}

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

SingleInherit sh = **new** SingleInherit();

sh.cashcounter();

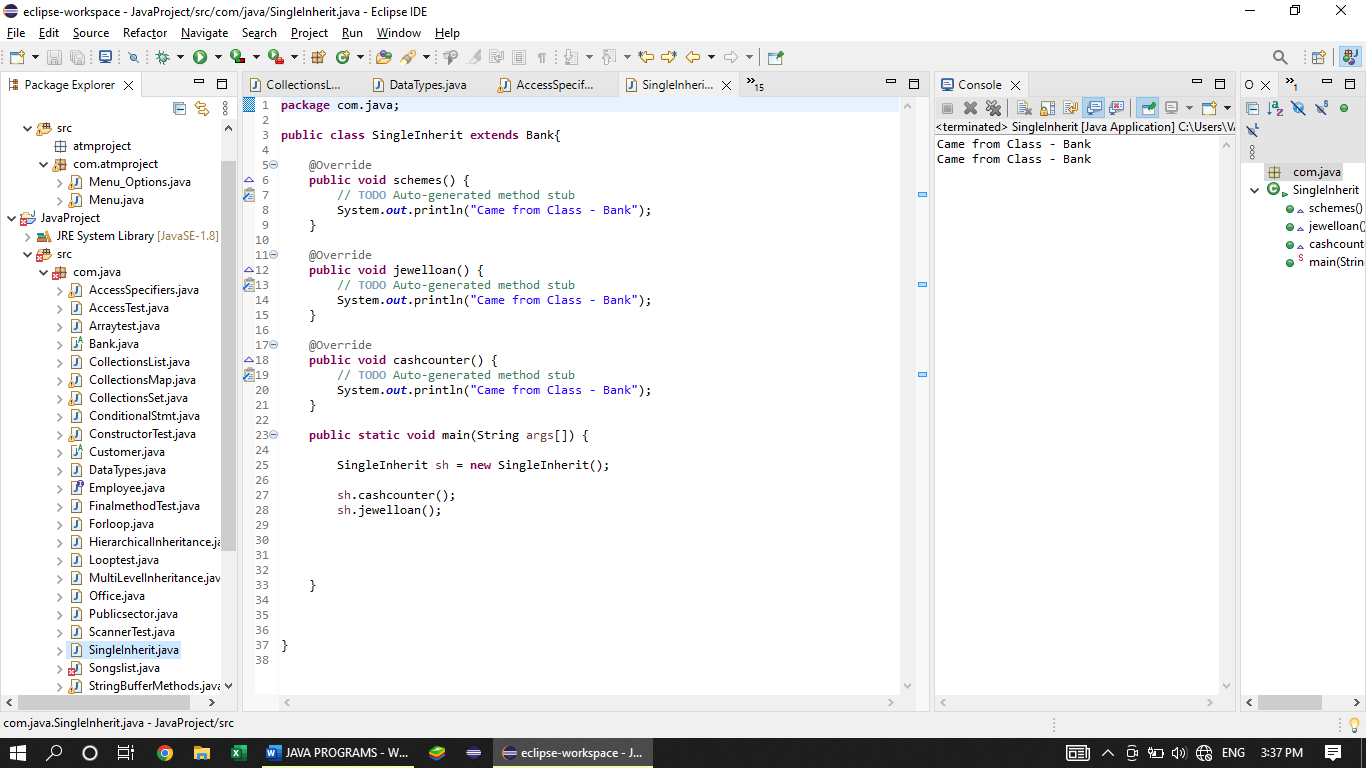
sh.jewelloan();

sh.withdraw();

}

}

**OUTPUT:**



1. **MULTILEVEL INHERITANCE**

**PARENT CLASS – BANK**

**package** com.java;

**public** **abstract** **class** Bank {

**public** **abstract** **void** schemes();

**public** **abstract** **void** jewelloan();

**public** **void** reception() {

System.***out***.println("How may i help u??");

}

**public** **abstract** **void** cashcounter();

}

**CHILD / SUB CLASS - CUSTOMER**

**package** com.java;

**public** **abstract** **class** Customer **extends** Bank{

**protected** **abstract** **void** withdraw();

**protected** **void** minbls() {

System.***out***.println("Your minimum balance should be maintained rs.1000!!!");

}

**public** **void** interestrate() {

System.***out***.println("your interest rate is 5% of your current balance ");

}

**public** **void** schemes() {

}

}

**SUB CHILD CLASS – MultiLevelInheritance**

**package** com.java;

**public** **class** MultiLevelInheritance **extends** Customer {

@Override

**public** **void** schemes() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Bank");

}

@Override

**public** **void** jewelloan() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Bank");

}

@Override

**public** **void** cashcounter() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Bank");

}

@Override

**protected** **void** withdraw() {

// **TODO** Auto-generated method stub

System.***out***.println("Came from Class - Customer");

}

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

MultiLevelInheritance Ml = **new** MultiLevelInheritance();

Ml.schemes();

Ml.jewelloan();

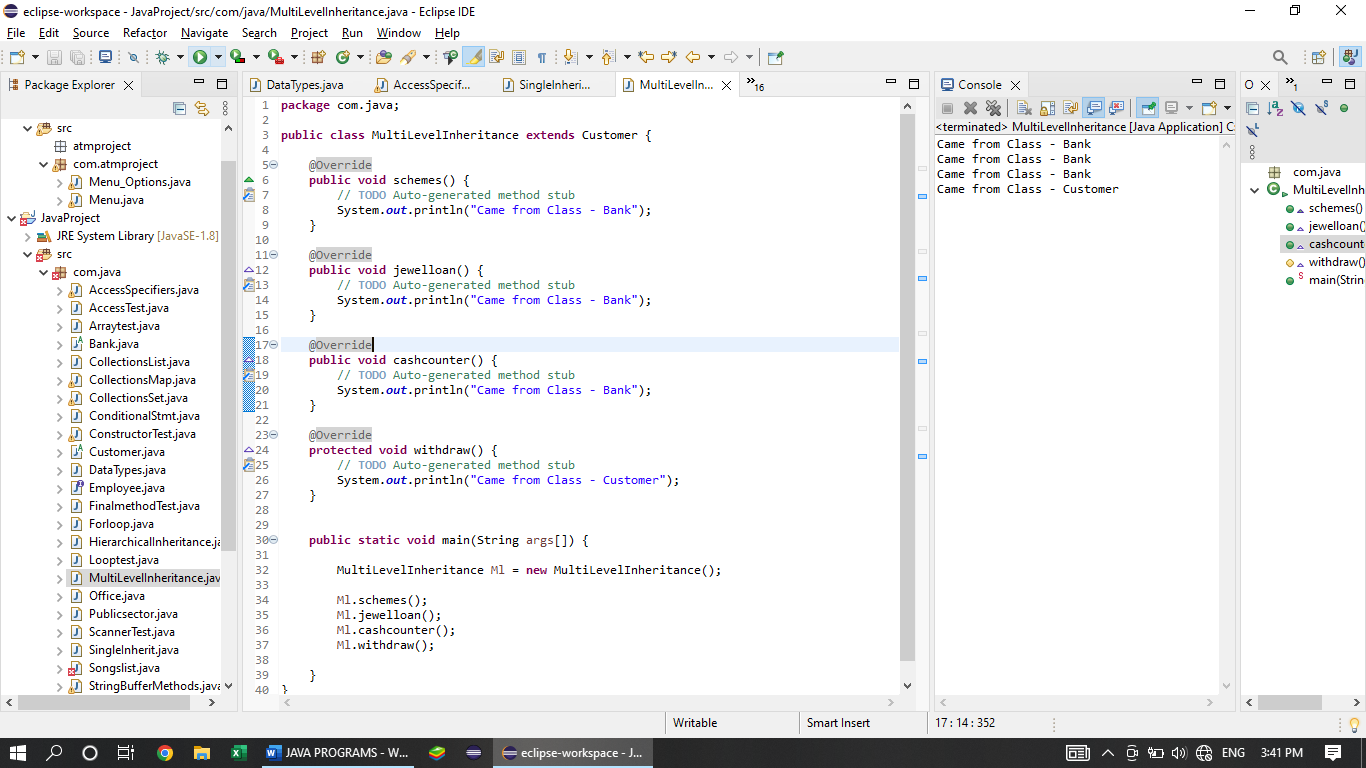
Ml.cashcounter();

Ml.withdraw();

}

}

**OUTPUT:**



1. **HIERARCHICAL INHERITANCE**

**package** com.java;

**public** **class** HierarchicalInheritance **extends** SingleInherit{

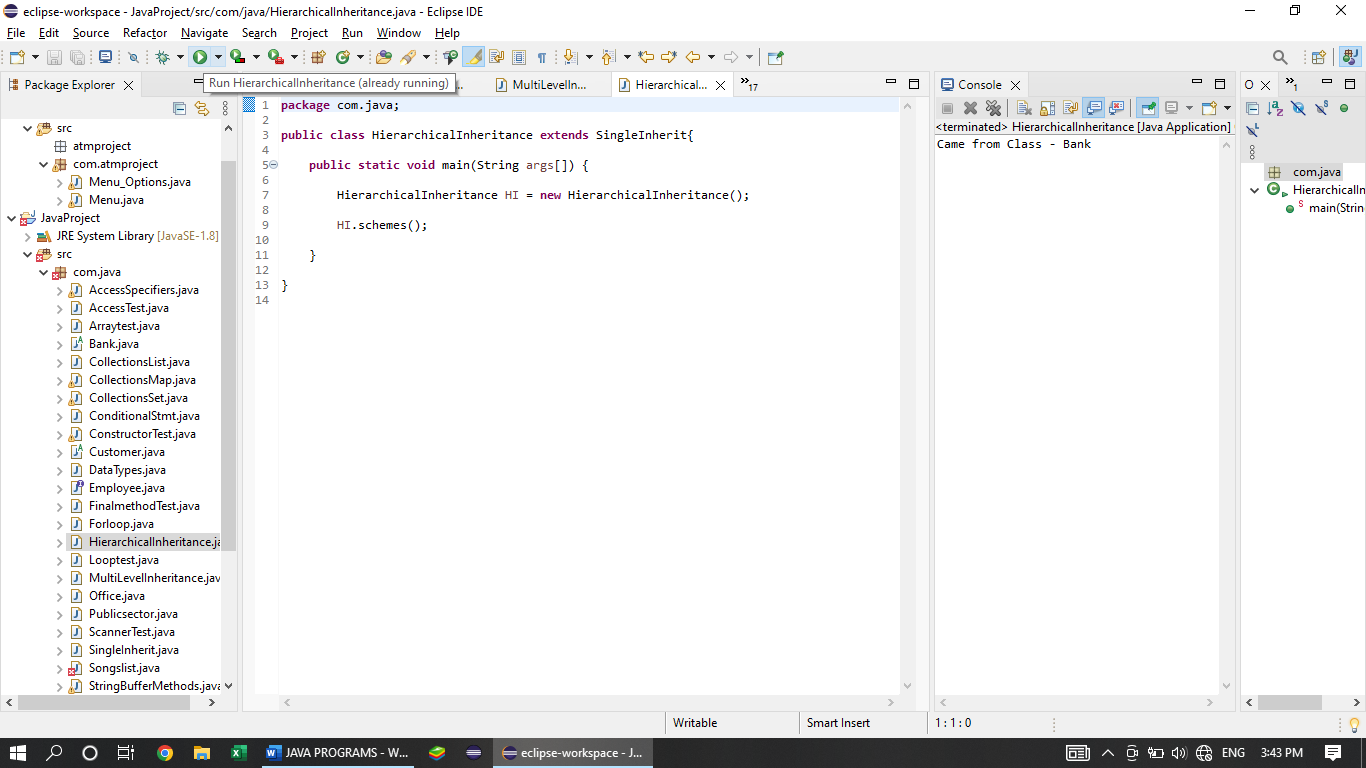
**public** **static** **void** main(String args[]) {

HierarchicalInheritance HI = **new** HierarchicalInheritance();

HI.schemes();

}

**OUTPUT:**



}

1. **ACCESS TEST**

**package** com.java;

**public** **class** AccessTest {

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

SingleInherit SI = **new** SingleInherit();

HierarchicalInheritance HI = **new** HierarchicalInheritance();

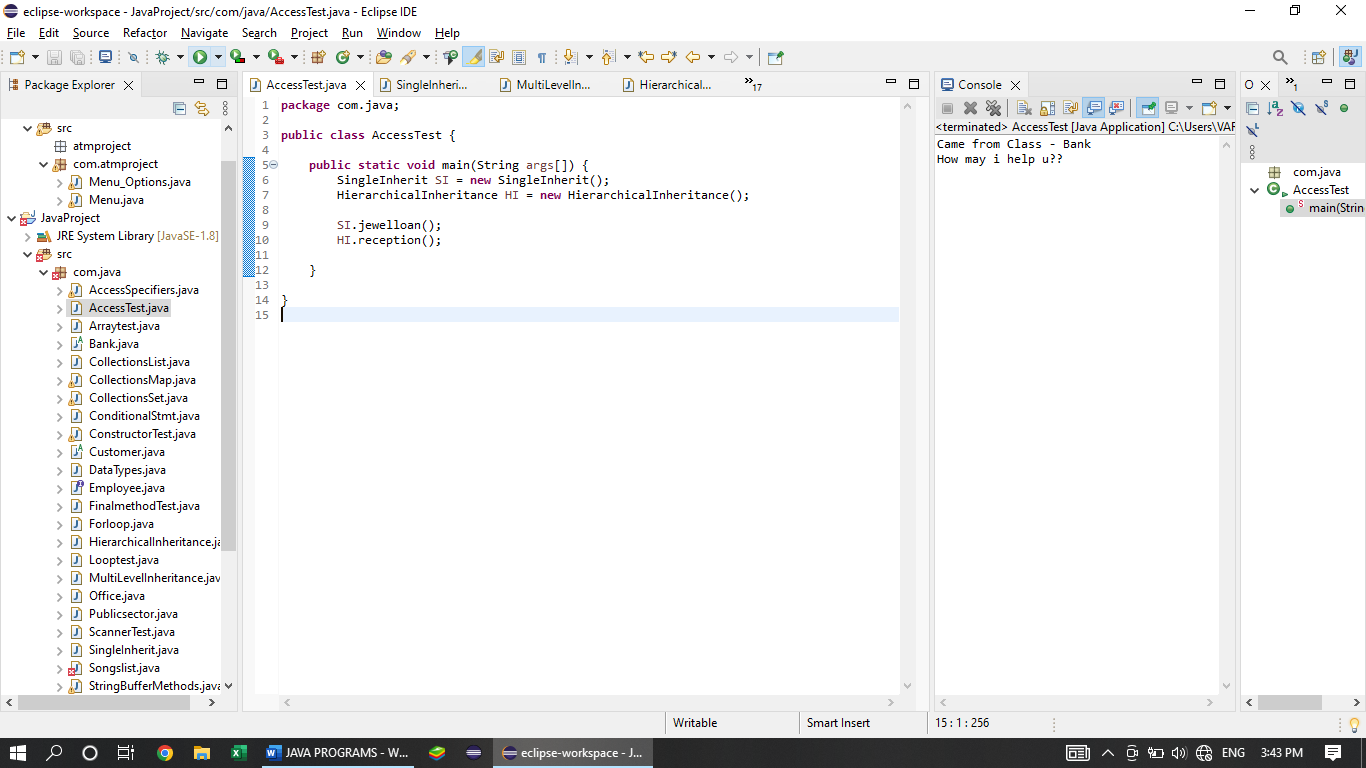
SI.jewelloan();

HI.reception();

}

}

**OUTPUT:**



1. **POLYMORPHISM**

**ABSTRACT CLASS**

**package** com.java;

**public** **abstract** **class** Bank {

**public** **abstract** **void** schemes();

**public** **abstract** **void** jewelloan();

**public** **void** reception() {

System.***out***.println("How may i help u??");

}

**public** **abstract** **void** cashcounter();

}

**EXTENDS CLASS**

**package** com.java;

**public** **abstract** **class** Customer **extends** Bank{

**protected** **abstract** **void** withdraw();

**protected** **void** minbls() {

System.***out***.println("Your minimum balance should be maintained rs.1000!!!");

}

**public** **void** interestrate() {

System.***out***.println("your interest rate is 5% of your current balance ");

}

**public** **void** schemes() {

}

}

**INTERFACE**

**package** com.java;

**public** **interface** Unemployee {

**public** **void** monthlydeposite();

**public** **void** governmentscheme();

**public** **void** minimumwithdraw();

}

**CLASS WITH EXTEDNS AND IMPLEMENTS**

**package** com.java;

**public** **class** Publicsector **extends** Customer **implements** Employee, Unemployee{

@Override

**public** **void** governmentscheme() {

// **TODO** Auto-generated method stub

System.***out***.println("The government announces the new schemes!!!");

}

@Override

**public** **void** minimumwithdraw() {

// **TODO** Auto-generated method stub

System.***out***.println("Your minimum withdraw amount is rs 1000");

}

@Override

**public** **void** monthlyincome() {

// **TODO** Auto-generated method stub

System.***out***.println("May i know your onthly income??");

}

@Override

**public** **void** ppfreduction() {

// **TODO** Auto-generated method stub

System.***out***.println("Your ppf reduction will be 2% of your monthly income.");

}

@Override

**public** **void** deposit() {

// **TODO** Auto-generated method stub

System.***out***.println("Please deposite your money here!!");

}

@Override

**public** **void** monthlydeposite() {

// **TODO** Auto-generated method stub

System.***out***.println("Monthly deposite amount:");

}

@Override

**public** **void** schemes() {

// **TODO** Auto-generated method stub

System.***out***.println("We give the scheme for various sectors...");

}

@Override

**public** **void** jewelloan() {

// **TODO** Auto-generated method stub

System.***out***.println("we are offering jewel loan at minimum interest");

}

@Override

**public** **void** cashcounter() {

// **TODO** Auto-generated method stub

System.***out***.println("You may collect your cash here!!!d");

}

@Override

**protected** **void** withdraw() {

// **TODO** Auto-generated method stub

System.***out***.println("Your withdrawn amount is:");

}

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

{

Publicsector P = **new** Publicsector();

P.governmentscheme();

P.minimumwithdraw();

P.monthlyincome();

P.cashcounter();

P.ppfreduction();

P.deposit();

P.jewelloan();

P.withdraw();

P.minbls();

P.schemes();

P.reception();

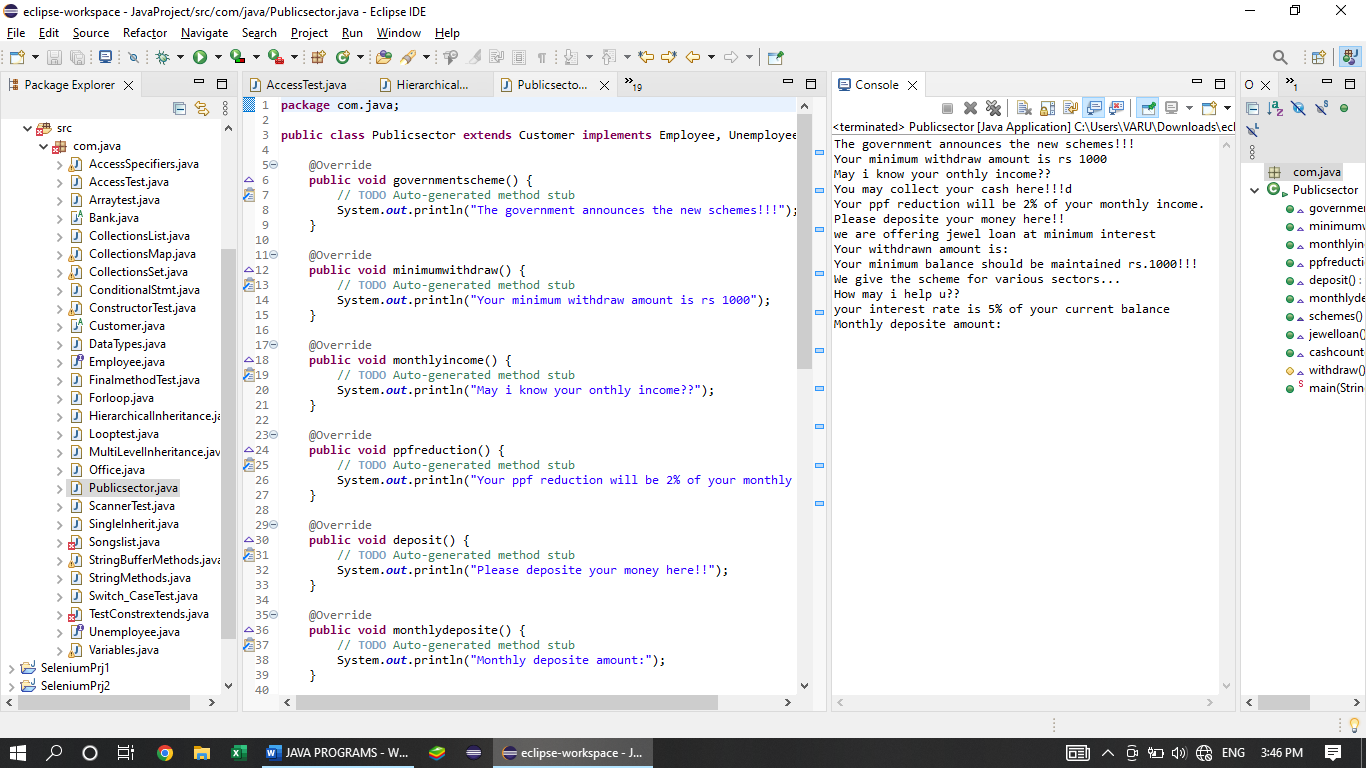
P.interestrate();

P.monthlydeposite();

}

}

**OUTPUT:**



1. **VARIABLES**

**package** com.java;

**public** **class** Variables {

String str = "Welcome to variables!!!"; // class variable declaration with initialization

**int** a; // class variable declaration without initialization.

**static** String *s*;

**final** **float** f= 2.5f;

**public** **void** classvar() {

// class variable can be accessed wherever.

// if value not assigned it will take default value

a=500;

**final** **char** g= 'g';

System.***out***.println("Your class variable in class is: "+str);

System.***out***.println("Your class variable inside the method is: "+ g);

}

**public** **void** localvar()

{

a=10;

// f=(float) 5.2; // final variable values cannot be changed

**final** **char** f= 'f';

**int** a =100;

String s = "value";

System.***out***.println("Your local variable in class is: "+ s);

System.***out***.println( "Your Local Variable is inside the method is:" + f );

}

**public** **void** staticvar() {

str = "static";

**final** **char** j='j';

System.***out***.println("Your static variable in class is:" + *s*);

System.***out***.println("Your static variable inside the method is: "+j);

}

**public** **void** finalvar() {

**final** **int** m = 99;

// s="java"; if we didnt declare the static variable it will take the default value

System.***out***.println("Your Final variable in class is:"+f);

System.***out***.println("Your final variable inside the method is:" + *s*);

// static int a=10; we cannot use the static keyword inside the method.

}

**public** **final** **void** finmeth() {

System.***out***.println("Final Method!!!");

}

**public** **final** **void** finmeth(**int** a, **char** c) {

System.***out***.println("Final Method with overloaded");

}

**public** **static** **void** stameth(**int** b,**double** d) {

System.***out***.println("static method with arguments");

}

**public** **static** **void** stamethd() {

System.***out***.println("Static method without arguments");

}

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

Variables L = **new** Variables();

L.classvar();

L.localvar();

L.staticvar();

L.finalvar();

L.finmeth();

L.finmeth(10,'f');

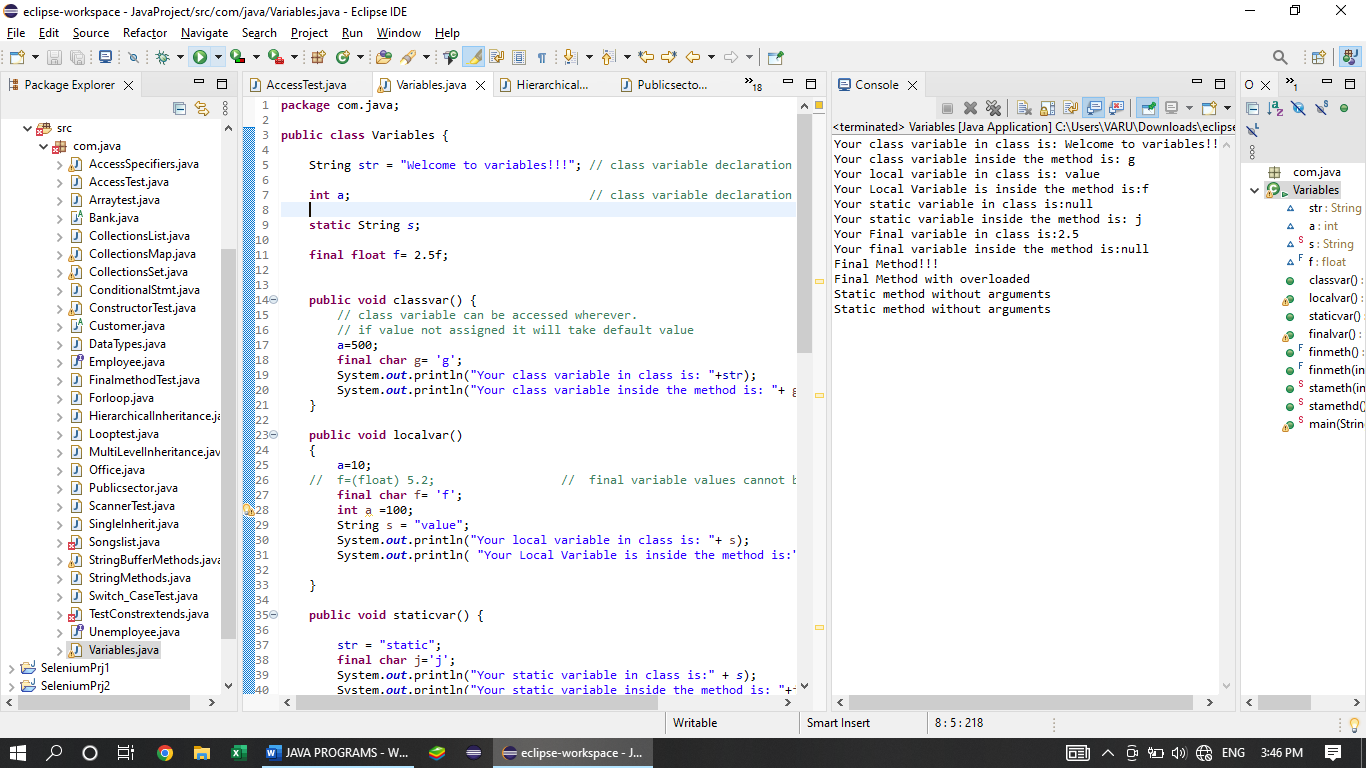
L.*stamethd*(); // static method obj can be created here too.. but we can create obj to another class

*stamethd*(); // static method can be called without creating an object

}

}

**OUTPUT:**



1. **CONSTRUCTOR**

**package** com.java;

**public** **class** ConstructorTest{

// public ConstructorTest(){ //default constructor

//

// }

**public** ConstructorTest(String name) {

System.***out***.println("Name is :" + name);

}

**public** ConstructorTest(**int** id, String dept) {

**this**();

System.***out***.println("Id and Department are :" + id +" "+dept);

}

**public** ConstructorTest() {

**this**("java");

System.***out***.println("Non parameterized Constructor");

**this**.method1();

**this**.method2();

}

**public** **void** method1() {

System.***out***.println("Test method 1");

}

**public** **void** method2() {

System.***out***.println("Test method 2");

}

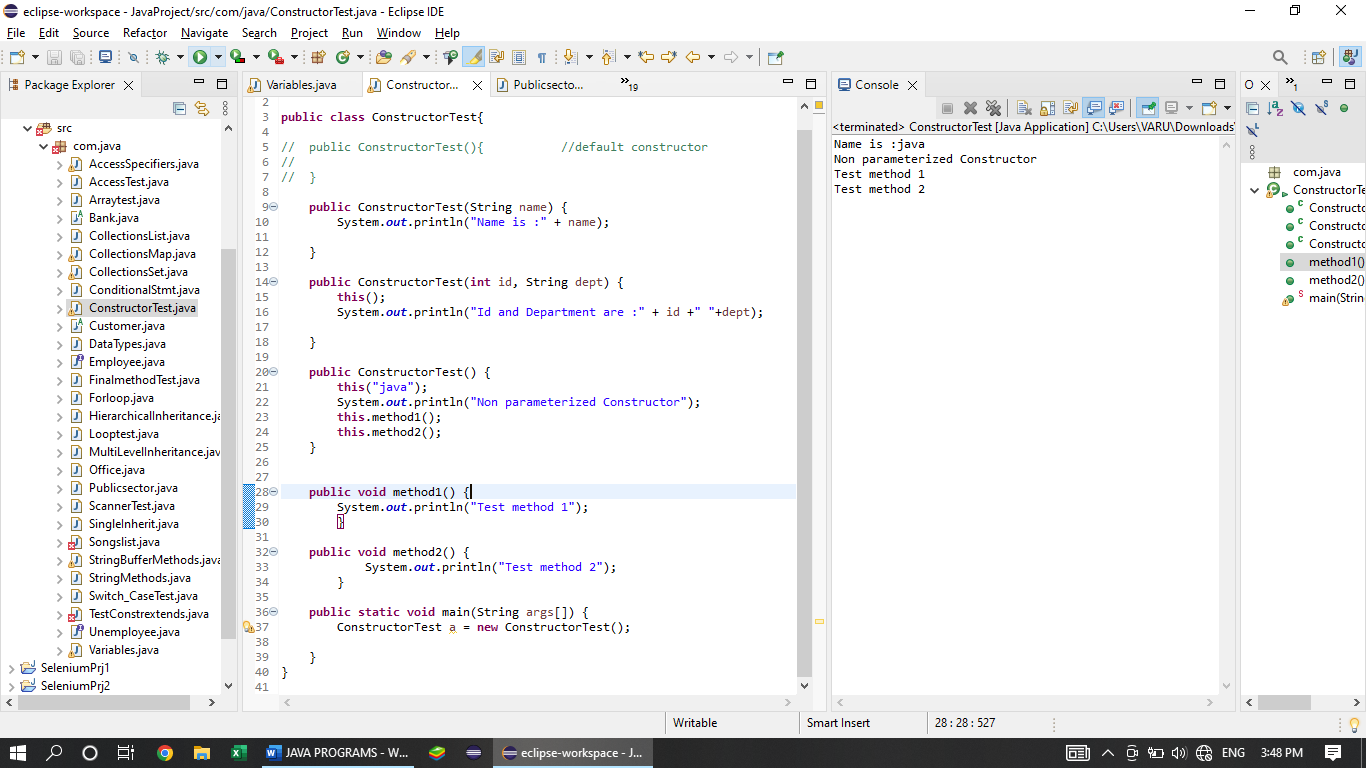
**public** **static** **void** main(String args[]) {

ConstructorTest a = **new** ConstructorTest(3,"java");

}

}

**OUTPUT:**



1. **SCANNER**

**package** com.java;

**import** java.util.Scanner;

**public** **class** ScannerTest {

Scanner s = **new** Scanner(System.***in***);

**public** **void** datatype() {

**int** i = s.nextInt();

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

String str = s.next();

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

**boolean** b = s.nextBoolean();

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

String str1= s.nextLine();

System.***out***.println(str1);

}

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

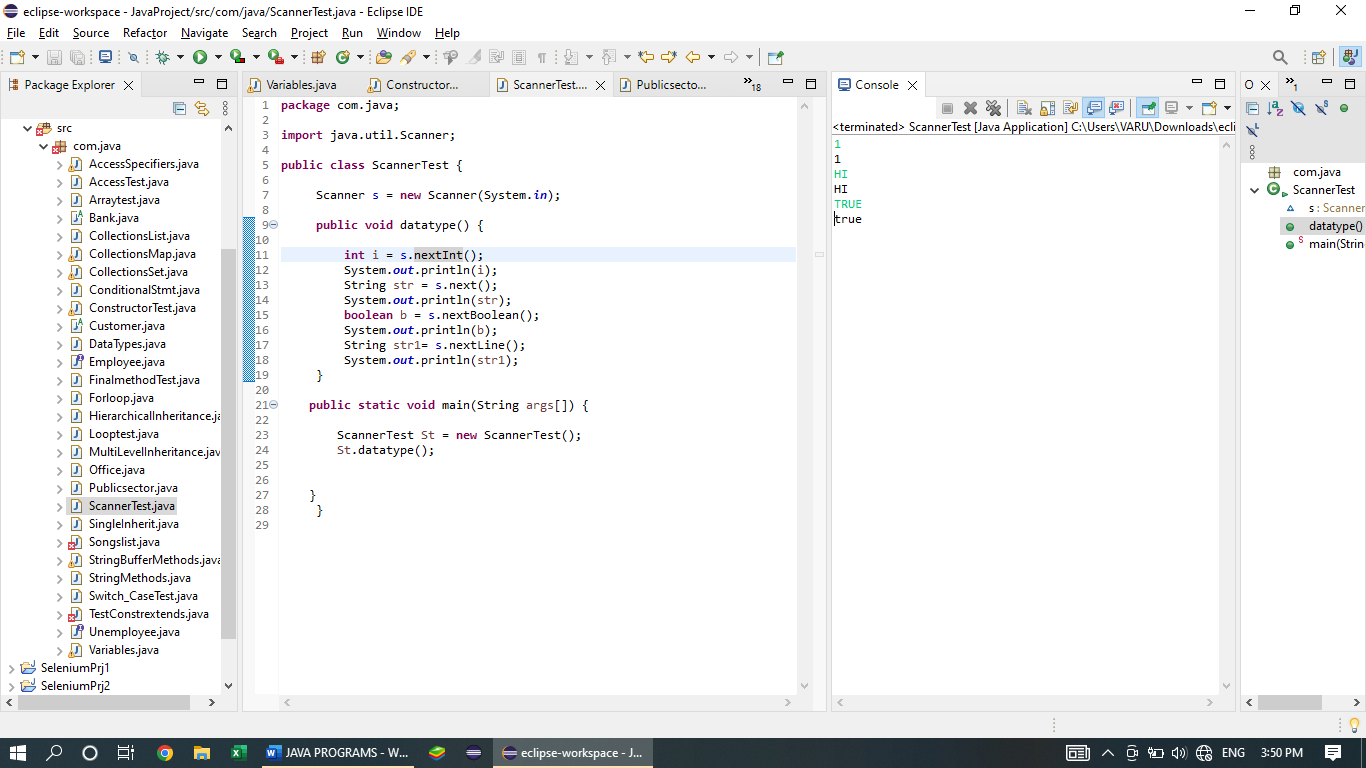
ScannerTest St = **new** ScannerTest();

St.datatype();

}

}

**OUTPUT:**



1. **CONTROL STATEMENTS**

**FOR AND WHILE LOOP**

**package** com.java;

**import** java.util.Scanner;

**public** **class** Looptest {

**public** Scanner s = **new** Scanner(System.***in***);

**public** **void** forloop() {

**int** i = s.nextInt();

**for**(i=0; i<=10; i++)

{

System.***out***.println("upward" + " " +i);

}

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

**for**( i=20; i>=10; i--)

{

System.***out***.print("downward" + " " +i+ " ");

}

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

}

**public** **void** whileloop()

{

**float** f = 2.5f;

**while**(f<=5.5) {

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

f++;

}

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

**while** (f>=0) {

System.***out***.print(f+" ");

f--;

}

}

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

Looptest L = **new** Looptest();

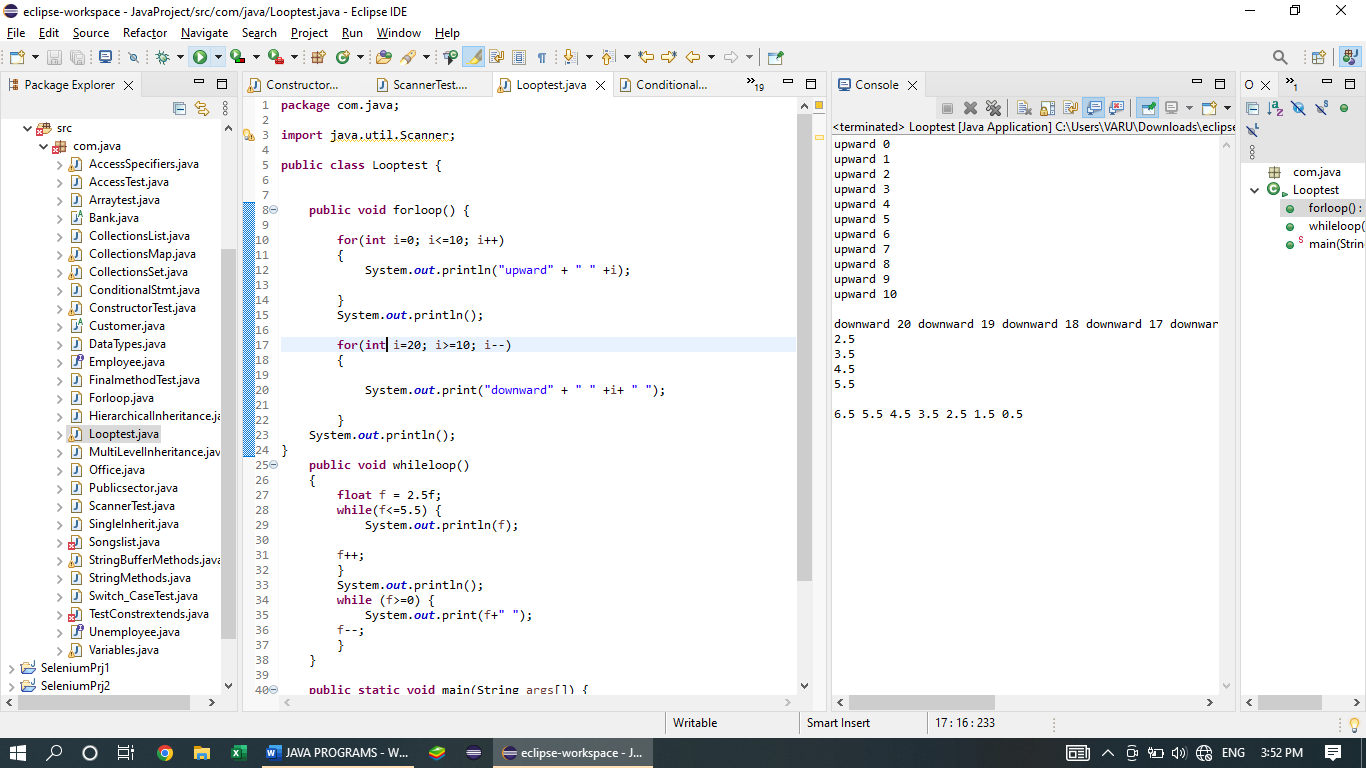
L.forloop();

L.whileloop();

}

}

**OUTPUT:**



**PATTERNS USING FOR LOOP**

**package** com.java;

**public** **class** Forloop {

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

//pattern using nested for loop

**for**(**int** i=10;i>=1;i--) {

**for**(**int** j=10;j>=i;j--) {

System.***out***.print(j);

}

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

}

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

**for**(**int** i=1;i<=10;i++) {

**for**(**int** j=10;j>=i;j--) {

System.***out***.print("\*");

}

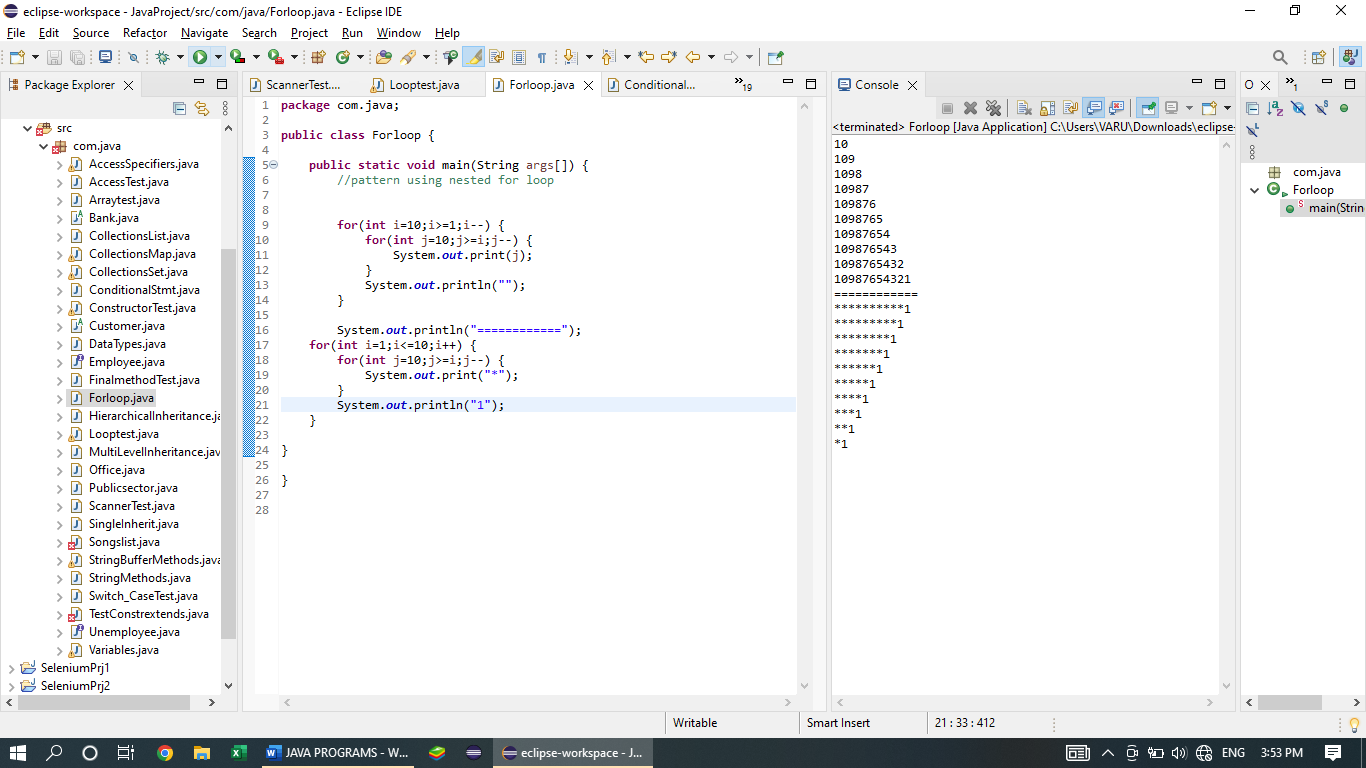
System.***out***.println("1");

}

}

}

**OUTPUT:**



1. **SWITCH STATEMENT**

**package** com.java;

**import** java.util.Scanner;

**public** **class** Switch\_CaseTest {

**static** Scanner *S* = **new** Scanner(System.***in***);

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

String Code = *S*.nextLine();

**switch** (Code) {

**case**("Bad request"):

System.***out***.println("400");

**break**;

**case**("Unautherized"):

System.***out***.println("401");

**break**;

**case**("Forbidden"):

System.***out***.println("403");

**break**;

**case**("Not Found"):

System.***out***.println("404");

**break**;

**case**("Internal server error"):

System.***out***.println("500");

**break**;

**default**:

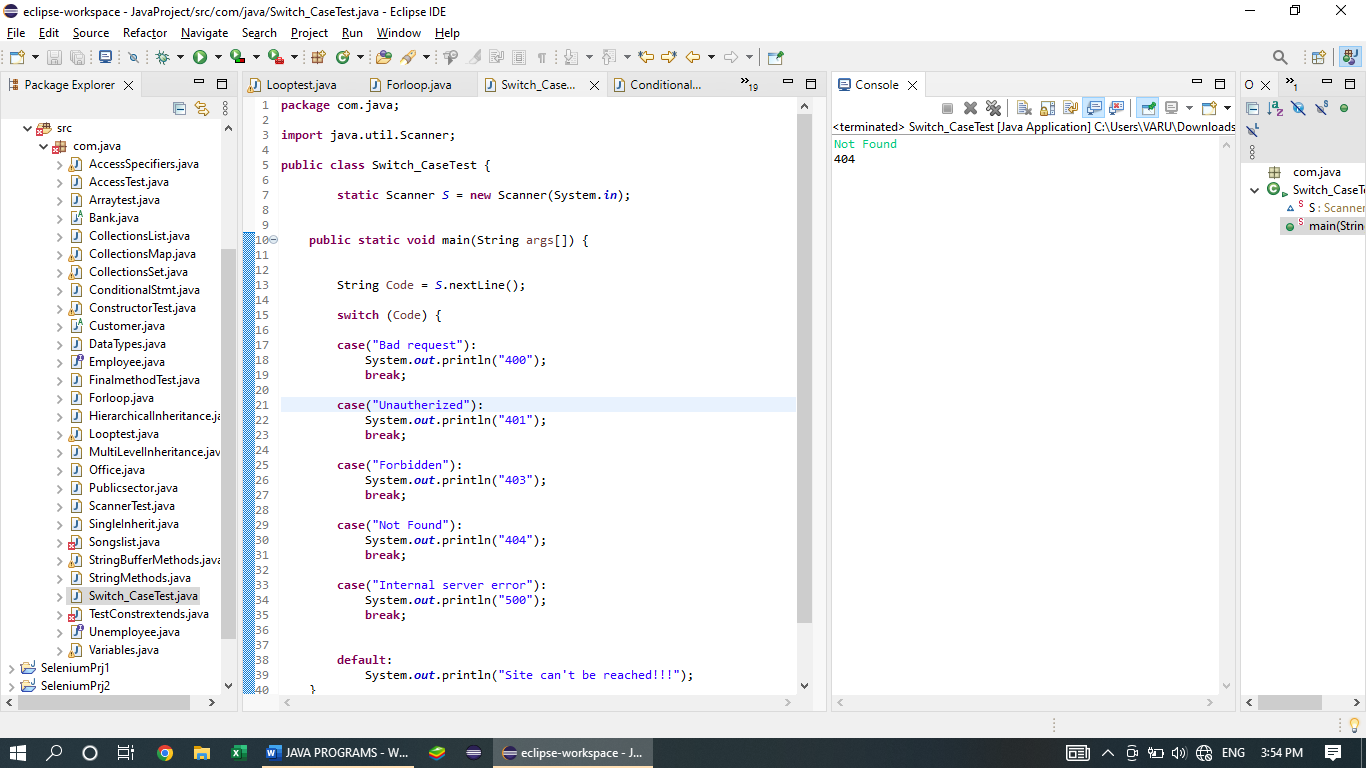
System.***out***.println("Site can't be reached!!!");

}

}

}

**OUTPUT:**



1. **CONDITIONAL STATEMENT**

**package** com.java;

**import** java.util.Scanner;

**public** **class** ConditionalStmt {

**public** Scanner s = **new** Scanner(System.***in***);

**public** **void** ifstmt() {

**int** age= s.nextInt();

**if**((age>=1)&&(age<=5)) {

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

}

**else** **if**((age>=6)&&(age<=17)) {

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

}

**else** **if**((age>=18)&&(age<=59)) {

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

}

**else** **if**((age>=60)&&(age<=100)) {

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

}

**else** {

System.***out***.println("invalid age!!!");

}

}

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

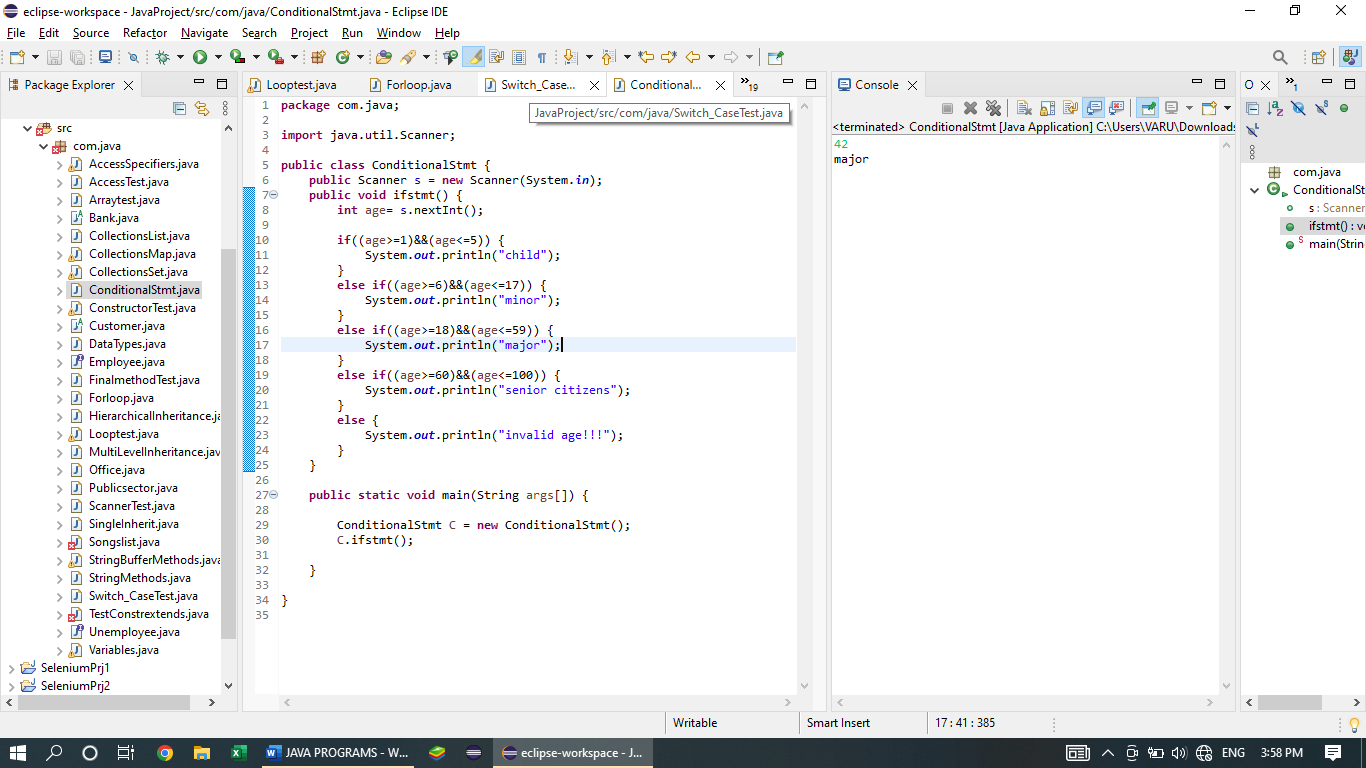
ConditionalStmt C = **new** ConditionalStmt();

C.ifstmt();

}

}

**OUTPUT:**



1. **ARRAY**

**package** com.java;

**public** **class** Arraytest {

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

String names [] = **new** String[6];

names [0] = "java";

names [1] = "selenium";

names [2] = "poly";

names [3] = "tag";

names [4] = "variable";

System.***out***.println(names.length);

System.***out***.println(names[3]);

**for**(**int** i = 0; i<names.length; i++) {

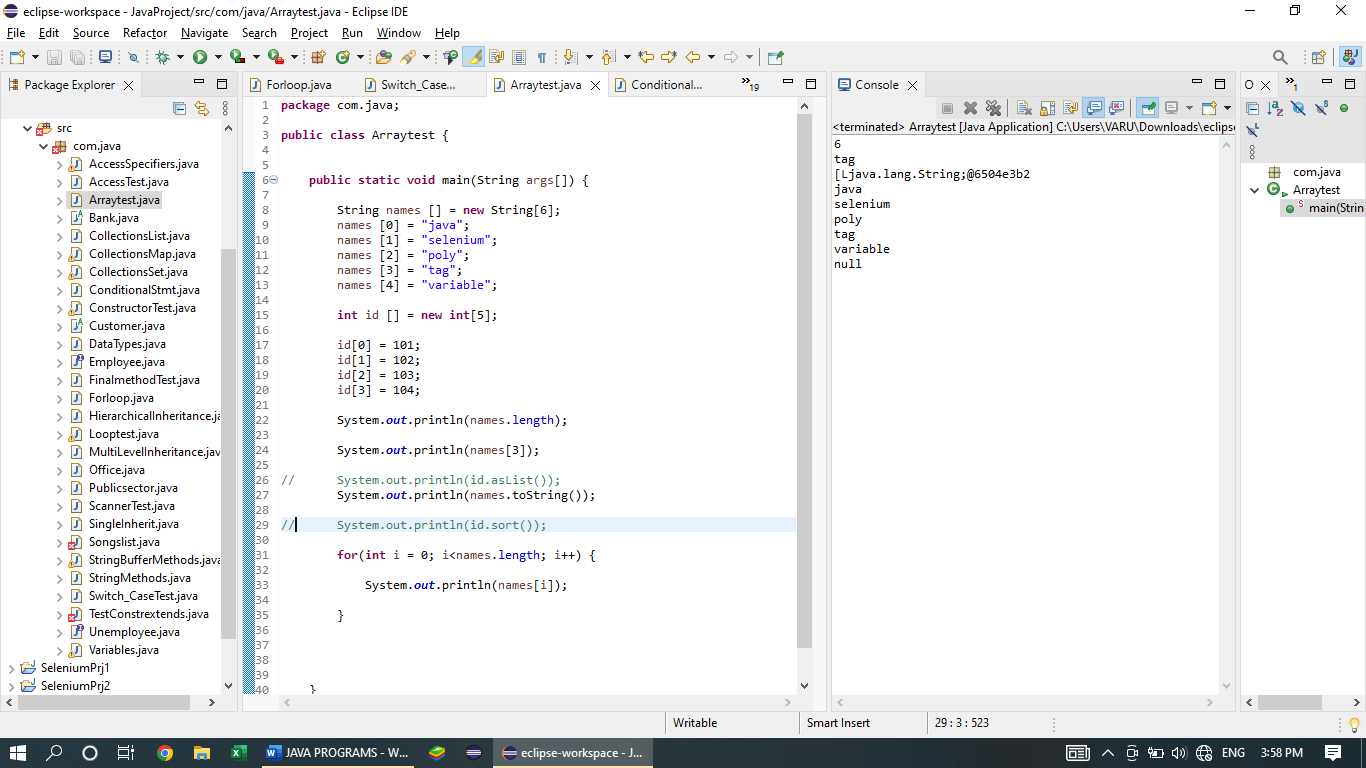
System.***out***.println(names[i]);

}

}

}

**OUTPUT:**



1. **STRING METHODS**

**package** com.java;

**public** **class** StringMethods {

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

String s1= "Akash";

String s2 = "Bala";

String s22 = "bala";

String s3 = "They Are good friends";

System.***out***.println(s1.length());

System.***out***.println(s1.equals(s3));

System.***out***.println(s1.equalsIgnoreCase(s22));

System.***out***.println(s1.concat(s1 + s2));

System.***out***.println(s1.toUpperCase());

System.***out***.println(s22.toLowerCase());

System.***out***.println(s3.charAt(6));

System.***out***.println(s3.indexOf('e'));

System.***out***.println(s2.lastIndexOf('a'));

System.***out***.println(s3.startsWith("are"));

System.***out***.println(s2.endsWith("la"));

System.***out***.println(s2.contains(s1));

System.***out***.println(s1.replace("la", "aa"));

System.***out***.println(s3.trim());

System.***out***.println(s3.substring(2,12));

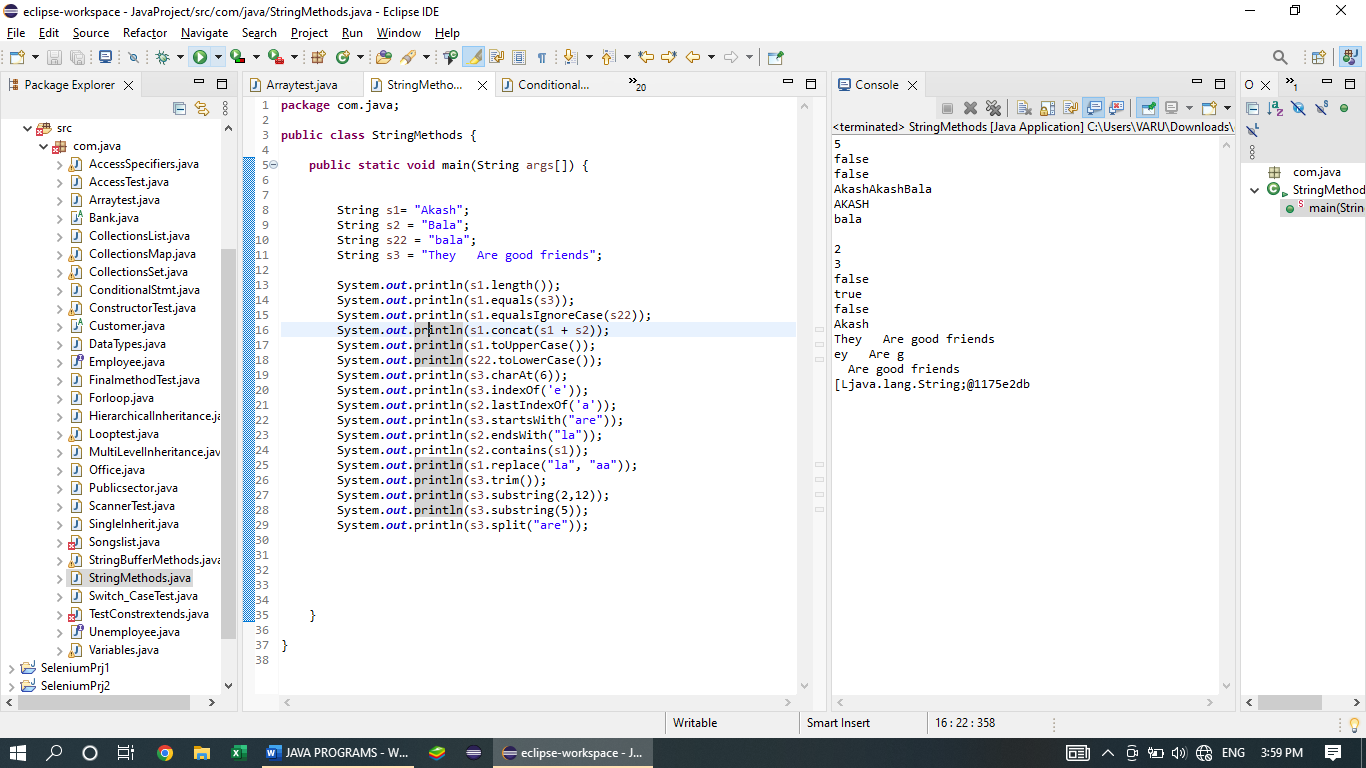
System.***out***.println(s3.substring(5));

System.***out***.println(s3.split("are"));

}

}

**OUTPUT:**



1. **COLLECTIONS - LIST**

package com.java;

import java.util.ArrayList;

import java.util.LinkedList;

import java.util.List;

import java.util.ListIterator;

import java.util.Vector;

public class CollectionsList {

public static void main(String args[]) {

List<Integer>sno = new ArrayList<>();

sno.add(01);

sno.add(02);

sno.add(03);

sno.add(04);

sno.add(05);

sno.add(06);

sno.add(07);

sno.add(8);

sno.add(9);

sno.add(10);

System.out.println(sno);

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

System.out.println(sno.contains(3));

System.out.println(sno.remove(1)); // remove the value in the given index

System.out.println(sno.addAll(sno));

ListIterator<Integer>iteration = sno.listIterator();

while(iteration.hasNext()) {

System.out.println(iteration.next());

}

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

while(iteration.hasPrevious()) {

System.out.println(iteration.previous());

}

List<String>name = new LinkedList<>();

name.add("AAA");

name.add("BBB");

name.add("CCC");

name.add("DDD");

name.add("EEE");

name.add("FFF");

name.add("GGG");

name.add("HHH");

name.add("WWW");

name.add("SSS");

// System.out.println(name.remove(1));

ListIterator<String>itr = name.listIterator();

while(itr.hasNext()) {

System.out.println(itr.next());

}

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

while(itr.hasPrevious()) {

System.out.println(itr.previous());

}

List<String>dept = new Vector<>();

name.add("AAA");

name.add("BBB");

name.add("CCC");

name.add("DDD");

name.add("EEE");

name.add("FFF");

name.add("GGG");

name.add("HHH");

name.add("WWW");

name.add("SSS");

System.out.println(dept.set(2, null));

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

System.out.println(dept.contains("ddd"));

// System.out.println(dept.remove(1)); // remove the value in the given index

// System.out.println(dept.addAll(dept));

// dept.clear(); // clear all values in the list

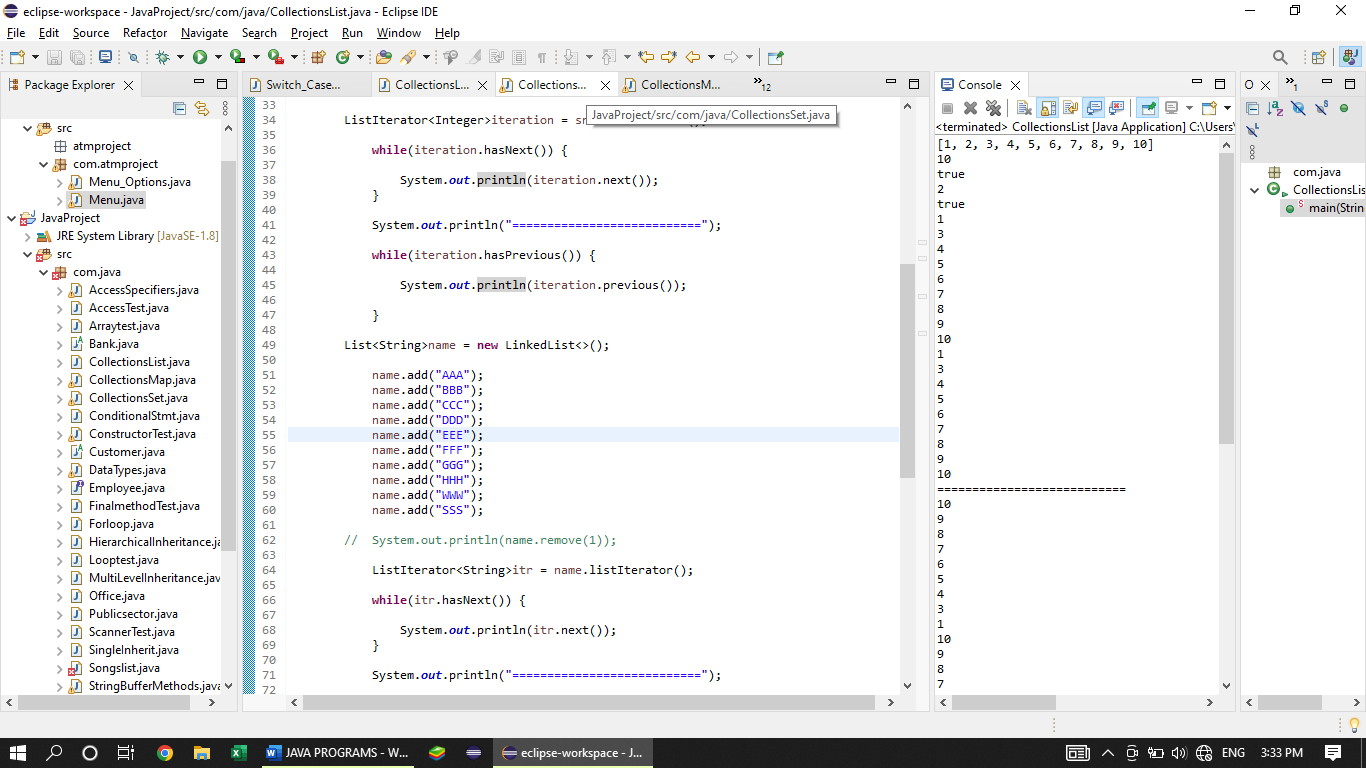
System.out.println(dept.removeAll(dept)); // remove the duplicate value

System.out.println(dept.retainAll(dept)); // retain the duplicate value only

}

}

**OUTPUT:**



1. **COLLECTIONS - SET**

package com.java;

import java.security.KeyStore.Entry;

import java.util.HashSet;

import java.util.LinkedHashSet;

import java.util.ListIterator;

import java.util.Set;

import java.util.TreeSet;

public class CollectionsSet {

public static void main(String args[]) {

Set<Integer>sno = new HashSet<>();

sno.add(01);

sno.add(02);

sno.add(03);

sno.add(04);

sno.add(05);

sno.add(06);

sno.add(07);

sno.add(8);

sno.add(9);

sno.add(10);

System.out.println(sno);

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

System.out.println(sno.contains(3));

System.out.println(sno.remove(1)); // remove the value in the given index

System.out.println(sno.addAll(sno));

Set<String>name = new LinkedHashSet<>();

name.add("AAA");

name.add("BBB");

name.add("CCC");

name.add("DDD");

name.add("EEE");

name.add("FFF");

name.add("GGG");

name.add("HHH");

name.add("WWW");

name.add("SSS");

// System.out.println(name.remove(1));

Set<String>dept = new TreeSet<>();

name.add("AAA");

name.add("BBB");

name.add("CCC");

name.add("DDD");

name.add("EEE");

name.add("FFF");

name.add("GGG");

name.add("HHH");

name.add("WWW");

name.add("SSS");

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

System.out.println(dept.contains("ddd"));

// System.out.println(dept.remove(1)); // remove the value in the given index

// System.out.println(dept.addAll(dept));

// dept.clear(); // clear all values in the list

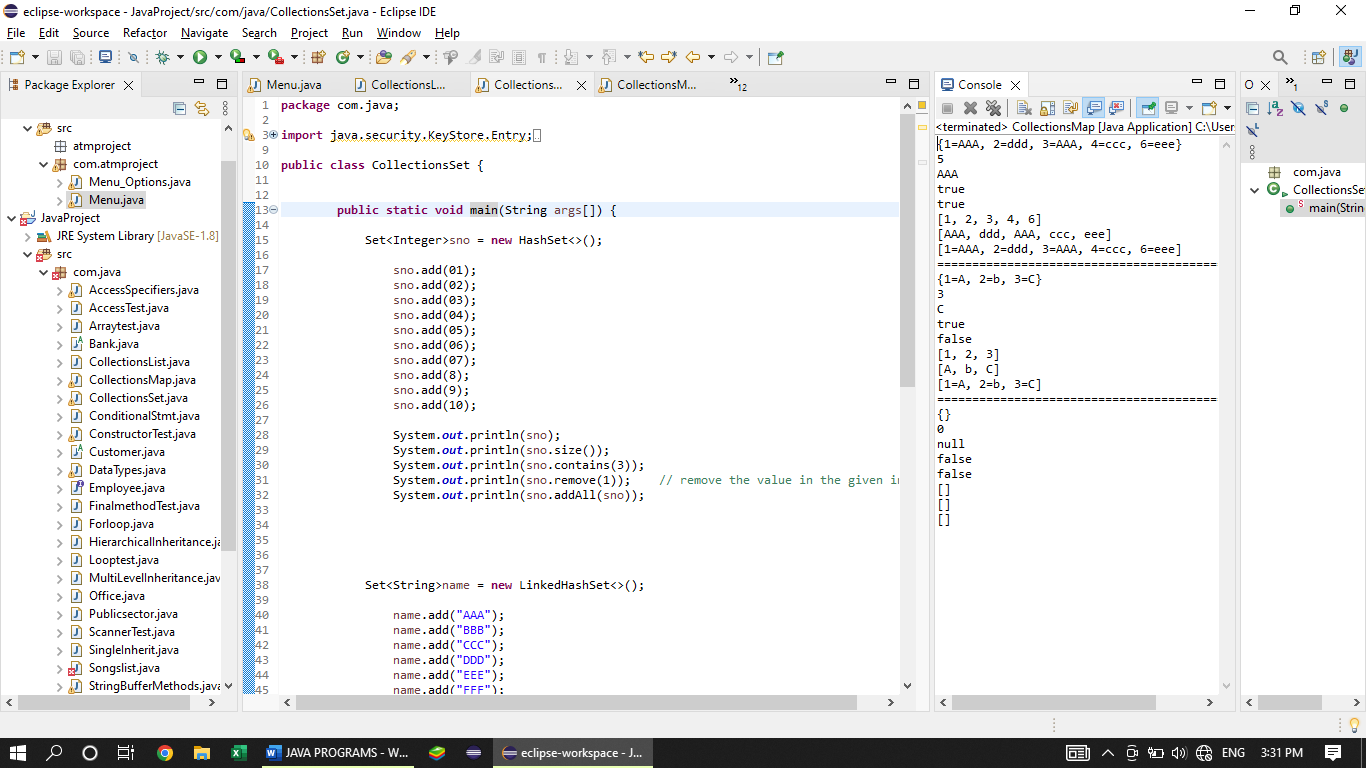
System.out.println(dept.removeAll(dept)); // remove the duplicate value

System.out.println(dept.retainAll(dept)); // retain the duplicate value only

}

}

**OUTPUT:**



1. **COLLECTIONS MAP**

package com.java;

import java.security.KeyStore.Entry;

import java.util.HashMap;

import java.util.HashSet;

import java.util.LinkedHashMap;

import java.util.LinkedHashSet;

import java.util.Map;

import java.util.Set;

import java.util.TreeMap;

import java.util.TreeSet;

public class CollectionsMap {

public static void main(String args[]) {

Map<Integer,String>sno = new HashMap<>();

sno.put(01,"AAA"); // put(key,value);

sno.put(02,"bbb");

sno.put(03,"AAA"); // duplicate value overloaded

sno.put(04,"ccc");

sno.put(02,"ddd"); // duplicate key overloaded

sno.put(06,"eee");

System.out.println(sno);

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

System.out.println(sno.get(3));

System.out.println(sno.containsKey(1)); // remove the value in the given index

System.out.println(sno.containsValue("ddd"));

System.out.println(sno.keySet());

System.out.println(sno.values());

System.out.println(sno.entrySet());

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

Map<Integer,String>name = new LinkedHashMap<>();

name.put(1,"A");

name.put(2,"b");

name.put(3,"C");

System.out.println(name);

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

System.out.println(name.get(3));

System.out.println(name.containsKey(1)); // remove the value in the given index

System.out.println(name.containsValue("ddd"));

System.out.println(name.keySet());

System.out.println(name.values());

System.out.println(name.entrySet());

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

Map<Integer,String>dept = new TreeMap<>();

name.put(1,"abc");

name.put(2,"def");

name.put(3,"ghi");

System.out.println(dept);

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

System.out.println(dept.get(3));

System.out.println(dept.containsKey(1)); // remove the value in the given index

System.out.println(dept.containsValue("ddd"));

System.out.println(dept.keySet());

System.out.println(dept.values());

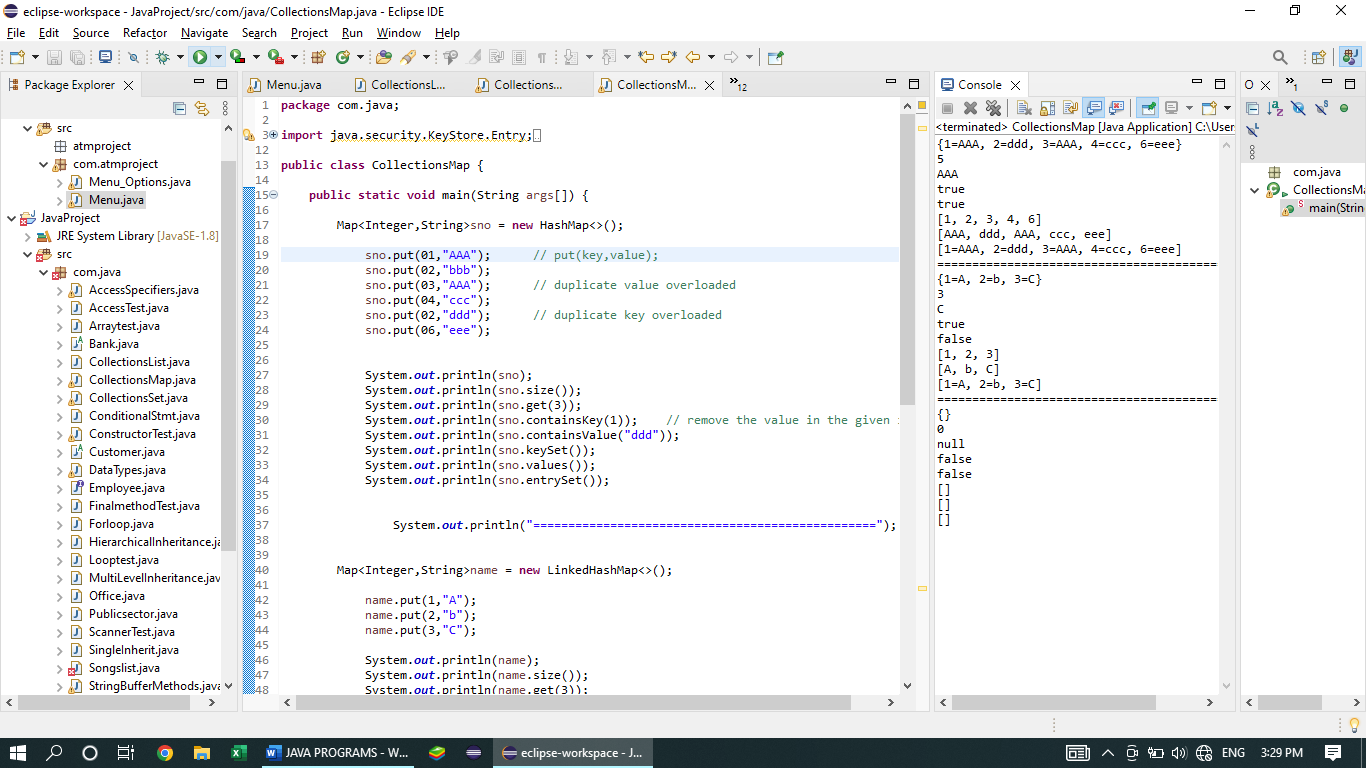
Set<java.util.Map.Entry<Integer,String>>entrySet = dept.entrySet();

System.out.println(dept.entrySet());

}

}

**OUTPUT:**



1. **MINI PROJECT**

**package** com.atmproject;

**import** java.util.Scanner;

**public** **class** Menu {

**static** Scanner *input*= **new** Scanner(System.***in***);

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

System.***out***.println("Enter your pin number: ");

**int** pin = 1234;

**int** passward = *input*.nextInt();

**int** balance = 10000;

**int** withdraw = 0;

**int** deposit = 0;

**if**(passward == pin)

{

System.***out***.println("Please Enter your name:");

String name = *input*.next();

System.***out***.println("Welcome "+ name);

**while** (**true**) {

System.***out***.println("Please select an Option below:");

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

System.***out***.println("Press 1 to deposit your money");

System.***out***.println("Press 2 to withdraw amount");

System.***out***.println("Press 3 to know your balance");

System.***out***.println("Press 4 to change your pin");

System.***out***.println("Press 5 to exit");

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

**int** option = *input*.nextInt();

**switch**(option) {

**case** 1:

**if**(option == 1) {

System.***out***.println("Please enter your deposited amount:");

**int** d = *input*.nextInt();

System.***out***.println("your Deposited amount is: "+d);

balance = balance+d;

System.***out***.println("Your current balance is: "+balance);

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

}

**break**;

**case** 2:

**if**(option==2) {

System.***out***.println("Please enter your withdraw amount:");

withdraw = *input*.nextInt();

**if** (withdraw > balance) {

System.***out***.println("Your balance is insufficient");

System.***out***.println("Try again!!!");

}**else** {

System.***out***.println("You have withdrawn Rs."+withdraw+ " ");

System.***out***.println("your current balance is: "+(balance - withdraw) );

}

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

}

**break**;

**case** 3:

**if**(option==3) {

**if**(balance>=0) {

System.***out***.println("your current balance is: "+ (balance-withdraw));

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

}**else** {

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

}

}

**break**;

**case** 4:

**if**(option==4) {

System.***out***.println(" Please enter your current pin: " );

**int** p = *input*.nextInt();

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

**if**(p==pin) {

System.***out***.println("Enter Your updated pin: ");

**int** pc = *input*.nextInt();

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

System.***out***.println("Your updated pin is: "+ pc);

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

}

**else** {

System.***out***.println("Enter the correct pin");

}

}

**break**;

**default**:

**if** (option == 5) {

System.***out***.println("Thank you for visiting");

**break**;

} }

}

}**else** {

System.***out***.println("Enter the correct pin");

}

}

}

**OUTPUT:**

