Java

Java is a popular programming language, created in 1995. It is owned by Oracle, and more than 3 billion devices run Java. It is used for: Mobile applications (specially Android apps).

METHOD**:**

A method is **a block of code which only runs when it is called**. You can pass data, known as parameters, into a method. Methods are used to perform certain actions, and they are also known as functions.

CODE:

**package** Practice;

**public** **class** method {

**static** **void** add(**int** a,**int** b){

**int** sum = a + b;

System.*out*.println(sum);

}

**public** **void** sub(**int** c,**int** d ){

**int** e = c - d;

System.*out*.println("sum is " + e);

}

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

*add*(5,5);

method minus = **new** method();

minus.sub(9,8);

}

}

OBJECT:

A Java object is a member (also called an instance) of a Java class. Each object has an identity, a behaviour and a state. The state of an object is stored in fields (variables), while methods (functions) display the object's behavior. Objects are created at runtime from templates, which are also known as classes.

For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

INHERITANCE:

Inheritance in Java is **a concept that acquires the properties from one class to other classes**; for example, the relationship between father and son. Inheritance in Java is a process of acquiring all the behaviours of a parent object.

For example, **a child inherits the traits of his/her parents**. With inheritance, we can reuse the fields and methods of the existing class.

CODE:

**package** Practice;

//multilevel inheritance

**class** animal {

String color = "white";

**int** age =5;

}

**class** cat **extends** animal{

String name = "kittu";

**public** **void** action(){

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

}

}

**class** dog **extends** cat{

String name = "snowy";

**void** act(){

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

}

}

**public** **class** inheri\_animal{

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

dog obj =**new** dog();

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

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

obj.action();

obj.act();

}

}

ABSTRACTION:

Data **abstraction** is the process of hiding certain details and showing only essential information to the user.  
Abstraction can be achieved with either **abstract classes** or [**interfaces**](https://www.w3schools.com/java/java_interface.asp)

CODE :

**package** Practice;

**import** java.util.\*;

**abstract** **class** vehical{

**abstract** **void** transport();

**void** color(){

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

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

String color = ab.nextLine();

System.*out*.println("color is " + color);

}

}

**class** truck **extends** vehical{

**void** transport(){

System.*out*.println("carries big loads");

}

}

**class** abstraction {

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

vehical ab = **new** truck();

ab.transport();

ab.color();

}

}

ENCAPSULATION:

The meaning of **Encapsulation**, is to make sure that "sensitive" data is hidden from users. To achieve this, you must:

* declare class variables/attributes as private
* provide public **get** and **set** methods to access and update the value of a private variable

CODE:

**package** Practice;

**class** enc {

**private** String designation;

**public** String getAme(){

**return** designation;

}

**public** **void** setAme(String newdesig){

**this**.designation = newdesig;

}

}

**public** **class** encaps{

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

enc obj = **new** enc();

obj.setAme("Manager");

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

}

}

CONSTRUCTOR

A constructor in Java is a **special method** that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes:

CODE

**package** Practice;

**public** **class** constructors {

//class name is ConstructorPgm.

**int** x;

**public** constructors() { //constructor name should be same as class name i.e ConstructorPgm

x=6;

}

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

constructors consObj=**new** constructors(); //once object is created then it is called automatically no need to call the obj .

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

}

}

COLLECTIONS

he **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

A Collection represents a single unit of objects, i.e., a group.

package sample;

import java.util.\*;

public class collection1{

public static void main(String[] args) {

Deque<String> deque = new ArrayDeque<String>();

deque.add("hii");

deque.add("all");

deque.add("good morning");

for (String st : deque) {

System.out.println(st);

HashSet<String> set=new HashSet<String>();

set.add("39");

set.add("65");

set.add("65");

set.add("39");

Iterator<String> itr=set.iterator();

while(itr.hasNext()){

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

List<String> list=new ArrayList<String>();

list.add("M");

list.add("A");

list.add("B");

list.add("G");

for(String letters:list)

System.out.println(letters);

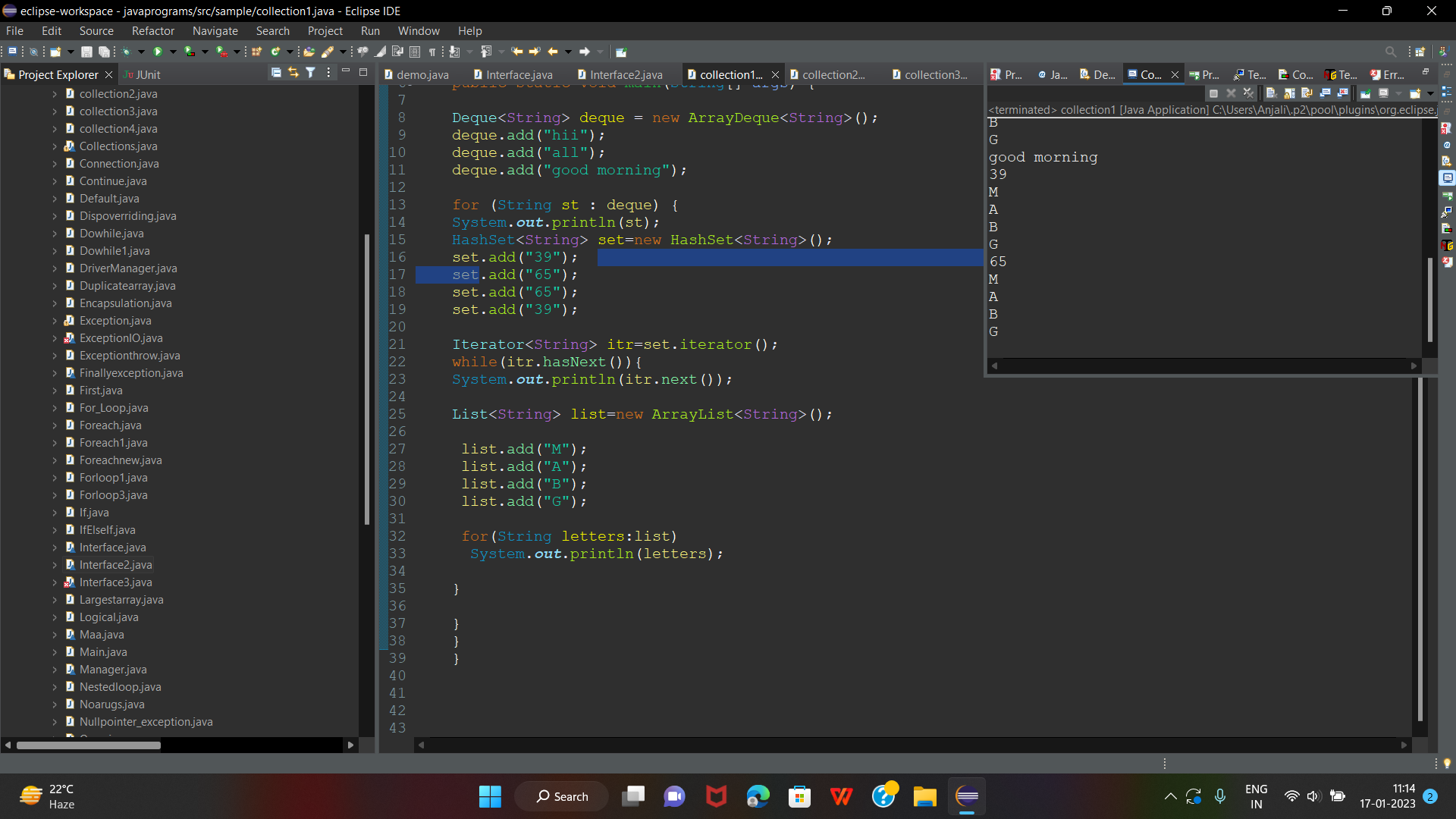
}

}

}

}

Output:



INTERFACES

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve*[*abstraction*](https://www.javatpoint.com/abstract-class-in-java). There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple [inheritance in Java](https://www.javatpoint.com/inheritance-in-java).

**package** Practice;

**import** java.util.\*;

**interface** subject{

**void** maths();

}

**interface** grade **extends** subject{

**void** firstclass();

}

**interface** result **extends** subject{

**void** pass();

}

**public** **class** interface3 **implements** subject,grade,result{

**int** marks;

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

**public** **void** pass(){

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

marks = sc.nextInt();

**if**(marks>60){

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

}

**else**{

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

}

}

**public** **void** firstclass()

{

//System.out.println("enter percent");

**int** percent =( marks/100)\*100;

//percent = sc.nextInt();

**if**(percent >=90){

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

}

**else**{

System.*out*.println("second or third class");

}

}

**public** **void** maths() {

System.*out*.println("u r good in maths");

}

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

interface3 ob= **new** interface3();

ob.pass();

ob.firstclass();

ob.maths();

}

}

Electricity bill in online:

package sample;

import java.util.Scanner;

public class electricitybillonline {

public static void main(String[] args) {

String name;

int mobile=970497;

int bill=500;

Scanner scanner=new Scanner(System.in);

System.out.println("enter mobile number");

int otp=scanner.nextInt();

if(otp==mobile)

{

System.out.println("enter your name");

name=scanner.next();

System.out.println("good morning"+name);

while(true)

{

System.out.println("enter your name");

int payment = 0;

switch(payment) {

case 1:

System.out.println("enter your mobile numebr");

break;

case 2:

System.out.println(" you can pay through phonepay");

int ShowAmount = scanner.nextInt();

System.out.println(" you have sufficient money to pay you can pay now");

break;

case 3:

System.out.println(" how much amount did you wnat to send");

int SendAmount = scanner.nextInt();

int balance = 0;

if(SendAmount>balance)

{

System.out.println(" if you are giving wrong password and otp");

}

else

{

System.out.println(" if you are giving correct number and otp it will pay");

break;

}

Object submit = null;

Object completed = null;

if(submit==completed)

{

System.out.println("thank you");

}

}

}

}

else {

System.out.println("wrong otp");

}

}

}

Output: