Java pgms qck:

**package** basicdt;

**public** **class** DT {

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

**byte** b =100; //min-128 mx 127 (-2p7to 2p7-1)

**short** s =257; //mn -32,768 mx 32,767(-2p15to 2p15-1)

**int** v = 123543;

**int** calc = -9876345;

**long** amountVal = 1234567891;

**float** intrestRate = 12.25f;

**double** sineVal = 12345.234d;

**boolean** flag = **true**;

**boolean** val = **false**;

**char** ch1 = 88; // code for X

**char** ch2 = 'Y';

System.*out*.println("byte Value = "+ b);

System.*out*.println("short Value = "+ s);

System.*out*.println("int Value = "+ v);

System.*out*.println("int second Value = "+ calc);

System.*out*.println("long Value = "+ amountVal);

System.*out*.println("float Value = "+ intrestRate);

System.*out*.println("double Value = "+ sineVal);

System.*out*.println("boolean Value = "+ flag);

System.*out*.println("boolean Value = "+ val);

System.*out*.println("char Value = "+ ch1);

System.*out*.println("char Value = "+ ch2);

}

}

**package** rough;

**public** **class** TestV

{// instance variables

**int** a = 10;

**int** b = 20;

// static variables

**static** **int** *c* = 30, *d* = 40;

// instance method

**int** m1(**int** a, **int** b)// local variables

{

System.*out*.println(a + "---" + b);

**return** 10;

}

// static method

**static** String m2(**boolean** b)// local variables

{

System.*out*.println(b);

**return** "ratan";

}

TestV(**int** a) // constructor with 1-arg

{

System.*out*.println("1-arg constructor");

}

TestV(**int** a, **int** b) {

System.*out*.println("2-arg constructor");

}

{

System.*out*.println("instance block-1");

} // instance block

{

System.*out*.println("instance block-2");

} // instance block

**static** {

System.*out*.println("static block-1");

} // static block

**static** {

System.*out*.println("static block-2");

} // static block

**public** **static** **void** main(String[] args) {// TestV object created with 1-arg

// constructor

TestV t1 = **new** TestV(10); // 1-arg constructor & instance blocks

// executed

// TestV object created with 2-arg constructor

TestV t2 = **new** TestV(100, 200); // 2-arg constructor & instance blocks

// executed

// printing instance variables by using Object name

System.*out*.println(t1.a);

System.*out*.println(t1.b);

// printing static variables by using class name

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

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

// instnace method calling by using object name

**int** x = t1.m1(1000, 2000);

System.*out*.println("m1() method return value:-" + x);

// printing return value static method calling by using class name

}

}

**package** rough;

**public** **class** Test

{

**int** a = 10; // instance variable

**static** **int** *b* = 20; // static variable

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

{

Test t = **new** Test();

System.*out*.println(t.a);// 10

System.*out*.println(t.*b*);// 20

t.a = 111;

t.*b* = 222;

System.*out*.println(t.a); // 111

System.*out*.println(t.*b*); // 222

Test t1 = **new** Test(); // 10 222

System.*out*.println(t1.a); // 10

System.*out*.println(t1.*b*);// 222

t1.a = 333;

t1.*b* = 444;

Test t2 = **new** Test();// 10 444

System.*out*.println(t2.a);// 10

System.*out*.println(t2.*b*);// 444

t2.*b* = 555;

Test t3 = **new** Test(); // 10 555

System.*out*.println(t3.a);// 10

System.*out*.println(t3.*b*);// 555

}

}

**package** rough;

**class** Test2

{

**int** x = 100; // instance variable(use this variable inside the class in

// multiple methods)

**static** **int** *y*=1000; // static variable

// (use this variable inside the class in multiple methods)

// instance method

**void** m1() // user defined method must called by user in main method

{

**boolean** b = **true**; // local variable declaration

System.*out*.println(b);

// //printing local variable

System.*out*.println(x);

// //printing instance variable

System.*out*.println(Test2.*y*); // printing static variable

}

// static method

**static** **void** m2() // user defined method must called by user in main method

{

**double** d = 10.5; // local variable

System.*out*.println(d); // printing local variable

Test2 t = **new** Test2();

System.*out*.println(t.x); // printing instance variable

System.*out*.println(Test2.*y*); // printing static variable

}

// static method

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

Test2 t = **new** Test2();

t.m1();

Test2.*m2*();

// instance method calling

// static method calling

}

}

**package** stringp;

**public** **class** Demo {

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

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

String ss = **new** String("a");

System.*out*.println("a".concat(ss));

String st = "Java is fun to learn";

String s1= "JAVA";

String s2= "Java";

String s3 = " Hello Java ";

System.*out*.println("Char at index 2(third position): " + st.charAt(2));

System.*out*.println("After Concat: "+ st.concat("-Enjoy-"));

System.*out*.println("Checking equals ignoring case: " +s2.equalsIgnoreCase(s1));

System.*out*.println("Checking equals with case: " +s2.equals(s1));

System.*out*.println("Checking Length: "+ st.length());

System.*out*.println("Replace function: "+

st.replace("fun", "easy"));

System.*out*.println("SubString of targetString: "+ st.substring(8));

System.*out*.println("SubString of targetString: "+ st.substring(8, 12));

System.*out*.println("Converting to lower case: "+ st.toLowerCase());

System.*out*.println("Converting to upper case: "+ st.toUpperCase());

System.*out*.println("Triming string: " + s3.trim());

System.*out*.println("searching s1 in targetString: " + st.contains(s1));

System.*out*.println("searching s2 in targetString: " + st.contains(s2));

**char** [] charArray = s2.toCharArray();

System.*out*.println("Size of char array: " + charArray.length);

System.*out*.println("Printing last element of array: " + charArray[3]);

}

}

**package** wrapperIntro;

**public** **class** ValueOfDemo {

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

Integer intWrapper = Integer.*valueOf*("12345");

//Converting from binary to decimal

Integer intWrapper2 = Integer.*valueOf*("11011", 2);

//Converting from hexadecimal to decimal

Integer intWrapper3 = Integer.*valueOf*("D", 16);

System.*out*.println("Value of intWrapper Object: "+ intWrapper);

System.*out*.println("Value of intWrapper2 Object: "+ intWrapper2);

System.*out*.println("Value of intWrapper3 Object: "+ intWrapper3);

System.*out*.println("Hex value of intWrapper: " + Integer.*toHexString*(intWrapper));

System.*out*.println("Binary Value of intWrapper2: "+ Integer.*toBinaryString*(intWrapper2));

}

}

**package** javarough;

**public** **class** Wc {

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

Integer in=10;

String s= "20";

String ss= "sASA";

**int** i=22;

System.*out*.println(in +" "+i+" "+ss+" "+ s);

//p.t to str

String s1= Integer.*toString*(i);

System.*out*.println("p.t to str"+s1);

//w.c to str

String s2= Integer.*toString*(in);

System.*out*.println("w.c to str"+s2);

//str to p.t

**int** n=Integer.*parseInt*(s);

System.*out*.println("str to p.t"+n);

//str to w.c

Integer x= Integer.*valueOf*(s);

System.*out*.println("str to w.c"+x);

//p.t to w.c

Integer k= i;

System.*out*.println("p.t to w.c"+k);

//w.c to p.t

**int** d = k;

System.*out*.println("w.c to p.t"+d);

//

}

}

**package** wrapperIntro;

**public** **class** WrapperDemo {

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

Integer intObj1 = **new** Integer (25);

Integer intObj2 = **new** Integer ("25");

Integer intObj3= **new** Integer (35);

//compareTo demo

System.*out*.println("Comparing using compareTo Obj1 and Obj2: " + intObj1.compareTo(intObj2));

System.*out*.println("Comparing using compareTo Obj1 and Obj3: " + intObj1.compareTo(intObj3));

//Equals demo

System.*out*.println("Comparing using equals Obj1 and Obj2: " + intObj1.equals(intObj2));

System.*out*.println("Comparing using equals Obj1 and Obj3: " + intObj1.equals(intObj3));

Float f1 = **new** Float("2.25f");

Float f2 = **new** Float("20.43f");

Float f3 = **new** Float(2.25f);

System.*out*.println("Comparing using compare f1 and f2: " +Float.*compare*(f1,f2));

System.*out*.println("Comparing using compare f1 and f3: " +Float.*compare*(f1,f3));

//Addition of Integer with Float

Float f = intObj1.floatValue() + f1;

System.*out*.println("Addition of intObj1 and f1: "+ intObj1 +"+" +f1+"=" +f );

}

}

**package** rough;

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

**public** **class** TestScanner {

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

{

Scanner s = **new** Scanner(System.*in*); // used to take dynamic input from

// keyboard

//

System.*out*.println("enter emp hobbies");

String ehobbies = s.nextLine();

System.*out*.println("enter emp no");

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

System.*out*.println("enter emp name");

String ename = s.next();

System.*out*.println("enter emp salary");

**float** esal = s.nextFloat();

System.*out*.println("\*\*\*\*\*emp details\*\*\*\*\*");

System.*out*.println("emp no----->" + eno);

System.*out*.println("emp name---->" + ename);

System.*out*.println("emp sal------>" + esal);

System.*out*.println("emp hobbies------>" + ehobbies);

s.close(); // used to close the stream

}

}

**package** io;

**import** java.io.\*;

**class** Test

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

{//Byte oriented channel

FileInputStream fis = **new** FileInputStream("abc.txt");

//read data from source file

FileOutputStream fos = **new** FileOutputStream("xyz.txt"); //write data to target file

**int** c; **while**((c=fis.read())!=-1)

{ System.*out*.print((**char**)c); fos.write(c);

}

System.*out*.println("read() & write operatoins are completed");

//stream closing operations

fis.close();

fos.close();

}

}

**package** ioexamples;

**import** java.io.File;

**import** java.util.Scanner;

**public** **class** File1 {

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

File file = **new** File("C://Users/Naresh/Desktop/waste.docx");

Scanner sc = **new** Scanner(file);

**while** (sc.hasNextLine() ){

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

}

}

}

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

// The name of the file to open.

String fileName = "C://Users/NARESH/Desktop/temp1.txt";

// This will reference one line at a time

String line = **null**;

**try** {

// FileReader reads text files in the default encoding.

FileReader fileReader =

**new** FileReader(fileName);

// Always wrap FileReader in BufferedReader.

BufferedReader bufferedReader =

**new** BufferedReader(fileReader);

**while**((line = bufferedReader.readLine()) != **null**) {

System.*out*.println(line);

}

// Always close files.

bufferedReader.close();

}

**catch**(FileNotFoundException ex) {

System.*out*.println(

"Unable to open file '" +

fileName + "'");

}

**catch**(IOException ex) {

System.*out*.println(

"Error reading file '"

+ fileName + "'");

// Or we could just do this:

// ex.printStackTrace();

}

}

}

package wasterough;

import java.io.FileOutputStream;

import java.io.IOException;

public class Filecreate {

public static void main(String[] args) {

// The name of the file to create.

String fileName = "temp1.txt";

try {

// Put some bytes in a buffer so we can

// write them. Usually this would be

// image data or something. Or it might

// be unicode text.

String bytes = "Hello theren";

byte[] buffer = bytes.getBytes();

FileOutputStream outputStream =

new FileOutputStream(fileName);

// write() writes as many bytes from the buffer

// as the length of the buffer. You can also

// use

// write(buffer, offset, length)

// if you want to write a specific number of

// bytes, or only part of the buffer.

outputStream.write(buffer);

// Always close files.

outputStream.close();

System.out.println("Wrote " + buffer.length +

" bytes");

}

catch(IOException ex) {

System.out.println(

"Error writing file '"

+ fileName + "'");

// Or we could just do this:

// ex.printStackTrace();

}

}

}

package rough;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

public class CollectionsDemo {

public static void main(String[] args) {

List<String> studentList = new ArrayList();

studentList.add("Neeraj");

studentList.add("Mahesh");

studentList.add("Armaan");

studentList.add("Preeti");

studentList.add("Sanjay");

studentList.add("Neeraj");

studentList.add("Zahir");

System.out.println("Original List " + studentList);

Collections.sort(studentList);

System.out.println("Sorted alphabetically List " + studentList);

Collections.reverse(studentList);

System.out.println("Reverse List " + studentList);

Collections.shuffle(studentList);

System.out.println("Shuffled List " + studentList);

System.out.println("Checking occurance of Neeraj: "

+ Collections.frequency(studentList, "Neeraj"));

}

}

**package** javarough;

**import** java.util.ArrayList;

**public** **class** ArrayListDemo {

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

{

ArrayList al = **new** ArrayList();

al.add(**new** Integer(10)); //boxing

al.add(20); //auto boxing (since jdk 1.5)

al.add(10); //duplicates allowed

al.add("amar");//heterogenous data allowed.

System.*out*.println(al);

}

}

**package** wasterough;

**public** **class** Threads **extends** Thread{

**public** **void** run()

{

**for**(**int** i=1 ; i<=500 ; i++)

{

System.*out*.print("i : "+i+"\t");

}

}

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

{

Threads t = **new** Threads(); //Thread Object

t.start(); //run() logic executes indepandantly

**for**(**int** j=1 ; j<=500 ; j++)

{

System.*out*.print("j : "+j+"\t");

}

}

**package** wasterough;

**import** java.util.Scanner;

**public** **class** In1 {

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

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

System.*out*.println("enter 2 numbers:");

**int** a= sc.nextInt();

**int** b= sc.nextInt();

**int** add = a+b;

**int** sub = a-b;

**float** mul = a\*b;

**double** div = a/b;

**int** mod = a%b;

**float** sqarea= a\*a;

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

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

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

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

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

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

}

}

**package** wasterough;

/\*\*

\* **@author** NARESH

\*

\*/

**public** **class** child **extends** parent{

child(){

**super**(10);

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

}

child(**int** a){

**this**();

System.*out*.println("hi iam 1 arg child");

}

child(**int** a,**int** b){

**this**(10);

System.*out*.println("hi iam 2 arg child");

}

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

child c = **new** child();

child c1 = **new** child(10);

child c2 = **new** child(10,20);

}

}

**package** main.java.com.in28minutes.java.arrays;

**import** java.util.Arrays;

**public** **class** ArrayExamples {

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

// Declare an Array. All below ways are legal.

**int** marks[]; // Not Readable

**int**[] runs; // Not Readable

**int**[] temperatures;// Recommended

// Declaration of an Array should not include size.

// int values[5];//Compilation Error!

// Declaring 2D ArrayExamples

**int**[][] matrix1; // Recommended

**int**[] matrix2[]; // Legal but not readable. Avoid.

// Creating an array

marks = **new** **int**[5]; // 5 is size of array

// Size of an array is mandatory to create an array

// marks = new int[];//COMPILER ERROR

// Once An Array is created, its size cannot be changed.

// Declaring and creating an array in same line

**int** marks2[] = **new** **int**[5];

// new Arrays are alway initialized with default values

System.***out***.println(marks2[0]);// 0

// Default Values

// byte,short,int,long-0

// float,double-0.0

// boolean false

// object-null

// Assigning values to array

marks[0] = 25;

marks[1] = 30;

marks[2] = 50;

marks[3] = 10;

marks[4] = 5;

// ArrayOnHeap.xls

// Note : Index of an array runs from 0 to length - 1

// Declare, Create and Initialize Array on same line

**int** marks3[] = { 25, 30, 50, 10, 5 };

// Leaving additional comma is not a problem

**int** marks4[] = { 25, 30, 50, 10, 5, };

// Default Values in Array

// numbers - 0 floating point - 0.0 Objects - null

// Length of an array : Property length

**int** length = marks.length;

// Printing a value from array

System.***out***.println(marks[2]);

// Looping around an array - Enhanced for loop

**for** (**int** mark : marks) {

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

}

// Fill array with same default value

Arrays.*fill*(marks, 100); // All array values will be 100

// Access 10th element when array has only length 5

// Runtime Exception : ArrayIndexOutOfBoundsException

// System.out.println(marks[10]);

// String Array: similar to int array.

String[] daysOfWeek = { "Sunday", "Monday", "Tuesday", "Wednesday",

"Thursday", "Friday", "Saturday" };

// Array can contain only values of same type.

// COMPILE ERROR!!

// int marks4[] = {10,15.0}; //10 is int 15.0 is float

// Cross assigment of primitive arrays is ILLEGAL

**int**[] ints = **new** **int**[5];

**short**[] shorts = **new** **short**[5];

// ints = shorts;//COMPILER ERROR

// ints = (int[])shorts;//COMPILER ERROR

// 2D Arrays

**int**[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 } };

**int**[][] matrixA = **new** **int**[5][6];

// First dimension is necessary to create a 2D Array

// Best way to visualize a 2D array is as an array of arrays

// ArrayOnHeap.xls

matrixA = **new** **int**[3][];// FINE

// matrixA = new int[][5];//COMPILER ERROR

// matrixA = new int[][];//COMPILER ERROR

// We can create a ragged 2D Array

matrixA[0] = **new** **int**[3];

matrixA[0] = **new** **int**[4];

matrixA[0] = **new** **int**[5];

// Above matrix has 2 rows 3 columns.

// Accessing an element from 2D array:

System.***out***.println(matrix[0][0]); // 1

System.***out***.println(matrix[1][2]); // 6

// Looping a 2D array:

**for** (**int**[] array : matrix) {

**for** (**int** number : array) {

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

}

}

// Printing a 1D Array

**int** marks5[] = { 25, 30, 50, 10, 5 };

System.***out***.println(marks5); // [I@6db3f829

System.***out***.println(Arrays.*toString*(marks5));// [25, 30, 50, 10, 5]

// Printing a 2D Array

**int**[][] matrix3 = { { 1, 2, 3 }, { 4, 5, 6 } };

System.***out***.println(matrix3); // [[I@1d5a0305

System.***out***.println(Arrays.*toString*(matrix3));

// [[I@6db3f829, [I@42698403]

System.***out***.println(Arrays.*deepToString*(matrix3));

// [[1, 2, 3], [4, 5, 6]]

// matrix3[0] is a 1D Array

System.***out***.println(matrix3[0]);// [I@86c347

System.***out***.println(Arrays.*toString*(matrix3[0]));// [1, 2, 3]

// Comparing Arrays

**int**[] numbers1 = { 1, 2, 3 };

**int**[] numbers2 = { 4, 5, 6 };

System.***out***.println(Arrays.*equals*(numbers1, numbers2)); // false

**int**[] numbers3 = { 1, 2, 3 };

System.***out***.println(Arrays.*equals*(numbers1, numbers3)); // true

// Sorting An Array

**int** rollNos[] = { 12, 5, 7, 9 };

Arrays.*sort*(rollNos);

System.***out***.println(Arrays.*toString*(rollNos));// [5, 7, 9, 12]

// Array of Objects(ArrayOnHeap.xls)

Person[] persons = **new** Person[3];

// Creating an array of Persons creates

// 4 Reference Variables to Person

// It does not create the Person Objects

// ArrayOnHeap.xls

System.***out***.println(persons[0]);// null

// to assign objects we would need to create them

persons[0] = **new** Person();

persons[1] = **new** Person();

persons[2] = **new** Person();

// Other way

// How may objects are created?

Person[] personsAgain = { **new** Person(), **new** Person(), **new** Person() };

// How may objects are created?

Person[][] persons2D = { { **new** Person(), **new** Person(), **new** Person() },

{ **new** Person(), **new** Person() } };

}

}

**class** Person {

**long** ssn;

String name;

}