Java

Git Useful things

First Time Code Checkout : git clone in tortoise git

Take the Update from repository: pull

To push the code to repository: commit & push

Reserved Keywords

class, public, static, void, int, float, double, short, char, boolean, byte, long, boolean , if, else, for, if else,volatile, new, final, finally, etc.

Data Types

3 types of Data Types

1. Numeric Data Types

2. Characters Data Types

3. Boolean Data Types

Numeric Data Types

1. byte
2. short
3. int
4. long
5. float
6. double

Characters Data Type

1. char

2. String

Boolean Data Types

1. Boolean

byte: -128 to 127

byte = 1 byte

1 byte = 8 bits

short = 2 bytes

short = -32768 to 32767

short = 16 bits

int = 4 bytes

int = -2147483648 to 2147483647

int = 32 bits

long = 8 bytes

long = **-9,223,372,036,854,775,808l (-263) to 9,223,372,036,854,775,807l**

long = 64 bits

float = 4 bytes

float = -2147483648.00f to 2147483647.00f

float = 32 bits

double = 8 bytes

double = **-9,223,372,036,854,775,808.00 (-263) to 9,223,372,036,854,775,807.00**

**double = 64 bits**

Characters Data Types

char

char = 2 bytes

char = 16 bits

String

boolean

boolean = 1 byte

main method

public static void main(String[] args) {

}

For compile : javac filename.java

For Run : java classname

26/11/2021

global variables vs local variables

global variables: class level variables we called as global variables

local variables: method level variables we called as local variables

ASCII Values

// local variables and conditional statements

If else or switch case

Syntax: if (condition) {

// statements

} else {

}

Find the number is even or odd

2,4,6,8,10…..

1,3,5,7…..

Find the given character is owel or not

a,e,I,o,u

27-11-2021

float percentage = 75.00f;

String name = “Naresh”;

If (percentage > 70) {

System.out.println(name+“ got first class”);

} else if (percentage > 60 && percentage < 70) {

System.out.println(name+“ got second class”);

} else if (percentage > 50 && percentage < 60) {

System.out.println(name+“ got third class”);

} else if (percentage > 40 && percentage < 50) {

System.out.println(name+“ got fourth class”);

} else {

System.out.println(name+“ got failed”);

}

If (percentage > 70) {

System.out.println(name+“ got first class”);

} if (percentage > 60 && percentage < 70) {

System.out.println(name+“ got second class”);

} if (percentage > 50 && percentage < 60) {

System.out.println(name+“ got third class”);

} if (percentage > 40 && percentage < 50) {

System.out.println(name+“ got fourth class”);

} if(percentage < 40){

System.out.println(name+“ got failed”);

}

Nested if else

Atm pin

int atmPin = 8545;

double balAmount = 50000.00;

double withDrawAmount = 40050.00;

if(atmPin == 8545) {

if(withDrawAmount % 100 == 0) {

if(balAmount > withDrawAmount) {

System.out.println(“Please Take Your Amount”);

} else {

System.out.println(“Insuffiecient Funds”);

}

} else {

System.out.println(“Invalid Amount”);

}

} else {

System.out.println(“Invalid Pin Please Try Again!”);

}

Task for 27-11-2021

1.Current Bill

2. owels (2 ways) if else if

3. results using all if

// nooutput

// nooutput

// 2/12/2021

If(true) {

System.out.println(“Naresh”);

}else If(true) {

System.out.println(“Naresh”);

} else If(true) {

System.out.println(“Naresh”);

}

Switch cases()

1 - > Sunday

2 -> Monday

3 -> Tuesday

4 -> WednesDay

5 -> ThursDay

6 -> Friday

7 -> Saturday

Syntax for switch case

Switch(“naresh”) {

Case “Sunday”: // statement

Case “Monday”: // statement

Case case: // statement

Default :

}

3/12/2021

Switch case and increment(++) and decrement(--) operators

1 to 10

Int num = 10;

Increment -> pre increment(++ num), post increment(num ++)

Decrement -> pre decrement(-- num), post decrement(num --)

Num++;

System.out.println(num++); //10

Looping Statements

Looping statement is nothing but a to execute the same code particular period of time.

We have 3 types of loops.

1. For loop
2. While loop
3. Do while loop

Syntax:

String name = “Naresh”;// initialization

Int I = 0; // initialization

1 2 4

for(initialization;condition;increment/decrement) {

3/5

}

30 < 1

31 <= 30

Int I;

for(int i=1; i<=100; i++) {

System.out.println(i); // 30,31,…50

}

1 is odd

2 is even

3 is odd

4 is even

99 is odd

100 is even

Prime number ? 3 is prime number

1 and number it self -> 2,3,5

Factorial -> 5 -> 1\*2\*3\*4\*5 = 120

6 -> 1\*2\*3\*4\*5\*6 = 720

7 -> 1\*2\*3\*4\*5\*6\*7 = 5040

Nested For Loop

For(int I = 1; i<6;i++) {

For(int j=1; j<6;j++) {

System.out.println(“Naresh”); // 10,10,25

}

}

While Loop

Syntax:

While(condition) {

}

I want to print the number 1 to 5

Int number = 1;

While(number < 6) {

System.out.println(number);

//Number++;

}

Do while - > is similar to while but there is slightly different.in while for the first time it won’t check the condition.

Syntax: do {

// statements

} while(condition);

Int I = 1;

do {

System.out.println(i);//1,2,3

i++;

} while(i < 11);

Tomorrow Session Agenda

14-12-2021

Logical Operators( && , ||,==,!=)

Ternary Operator ->

Condition ? statement : statement;

If(num== 10) System.out.println(“Given Number is Ten”);

Else System.out.println(“Given Number is Not Ten”);

1 2 3

Num == 10 ? “Given Number is Ten” : “Given Number is Not Ten”;

Variable vs Array

Int age;

Int[] ages = new int[5]; // one way of creating array by giving the size

// Initialising the array

Ages[0] = 10;

Ages[1] = 20;

Ages[2] = 30;

Ages[3] = 40;

Ages[4] = 50;

ages[6] = 10;

// Printing the array

System.out.println(ages[9]);

System.out.println(ages[8]);

System.out.println(ages[7]);

System.out.println(ages[6]);

System.out.println(ages[5]);

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

System.out.println(ages[3]);

System.out.println(ages[2]);

System.out.println(ages[1]);

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

15-12-2021

No New Concept examples on Arrays

16-12-2021

WAP to remove the duplicate numbers in array.

Sample: {1,2,3,1,2,5,6,3};

The above program has written and committed to git.

Tasks ->

Write the program for ascending order for given char array

Write the program to remove the duplicate characters in a given array

17-12-2021

WAP using float Array ascending and descending order

Multi Dimensional Array

Examples are discussed in the class

Task

swap the rows in multi dimensional array

print the sum of 1 to 100 factorial -> 1+2+6+24+120+720+5040….= sum

Find the counter of given array

Int[] arr = {0,1,3,2,12}; -> 0, 1,2,-2,6

Disarium Number

153 -> 1 + 5\*5 + 3\*3\*3 = 1+25+27 = 53

175 -> 1+ 7\*7 + 5\*5\*5 = 1+49+125 = 175

1254 -> 1+2\*2 + 5\*5\*5 + 4\*4\*4\*4 = 1+4+125+256 = 386

20-12-2021

OOPS -> Object Oriented Programming Structure

OOPS principles ->

1. Class
2. Object
3. Inheritance
4. Abstraction
5. PolyMorphism
6. Encapsulation

Class

1. To transfer the data
2. To Write the Business Logics

To represent group of objects we can create a class.

Class is a blueprint.

Class is a user defined data type;

Class is not a physically Exist;

Class Men {

String name;

String color;

Float weight;

Float height;

Int age;

}

Class Animal {

String animalType;

}

Class Women {

String name;

String color;

Float weight;

Float height;

Int age;

}

Class Human {

String name;

String color;

Char gender;

Float weight;

Float height;

Int age;

}

Class Bank {

String name=”RBI”;

String branch;

String city;

String IFSC;

}

Class Train {

String name;

Int number;

String startFrom;

String end;

}

Class Mobile {

String brand;

String IMEI;

String color;

Float cost;

}

Class Bike {

String brand;

String cc;

Float cost;

String name;

}

Class ElectionParty {

String partyName;

String president;

String symbol;

String founder;

String president;

}

Class State {

String name;

String cm;

String capital;

}

Class Village {

String villageName ;

String mandal;

String district;

String state;

Int pincode;

Int population;

}

2.Object

Object will be a physically exist.

Object is a implementation of class.

Village v = new Village();

Village v2 = new Village();

21-12-2021

Package and import

Class Fruit {

String name;

String taste;

String color;

}

Fruit f = new Fruit();

f.name = “Mango”;

f.taste = “Sweet”;

f.color = “Yellow”;

System.out.println(f.name);

System.out.println(f.taste);

System.out.println(f.color);

JVM Architecture or JVM Memory Management



Heap Area -> all the objects will be stored in heap area.

Method Area -> all class names and method names and meta data will be stored in method area.

Stack Area -> Method call stacks will be stored in Stack Area.

22-12-2021

Access Modifiers

Method creation

Access Modifiers

1. public
2. private
3. protected
4. default

Class Employee { public class Address {

}

}

Method syntax

Access modifiers returntype methodname(input paramaeters) {

}

Public long findFactorial(long input) {

Return fact;

}

Public void findFactorialAndPrint(long input) {

System.out.println(fact);

}

Public Boolean isPolindrom(long input) {

Return true/false;

}

Public void findAndPrintPolindrom(long input) {

System.out.println(true);

}

23-12-2021

Arithmetic Operations explained in class

WAP to find given number is Abundant or Deficient or Perfect Number.

6 -> 1+2+3 = 6 Perfect

12 -> 1+2+3+4+6 = 16 Abudant

14 -> 1+2+7 = 9 deficient

24-12-2021

Task For Today

{1,5,6,8,9,10} -> 9 -> 0,3

{8,9,10,25,,30,35} -> 40 -> 2, 4

Static vs not static

Static variable

Static method

Static block

Static class

25-12-2021

class StaticExample

{

static int num = 10;

static {

System.out.println(num);

num = 20;

}

static {

System.out.println(num);

num = 30;

}

public static void main(String[] args)

{

System.out.println(num);

}

}

Static method vs not static methods

Static method -> static methods will load at the time of class loading time, so If we want call the static methods we can call by using class name.

Non-static method -> non static methods will load at the time of object creation time, so if we want call the non-static methods we can by using object reference.