1.Git

2. Core Java ->

2.1 -> Conditional Statements i.e if else if else

Current Bill Generation

10 units -> 10 per unit = 10\*10 =100

15 units -> 15 per unit

20 units -> 20 per unit

Above -> 30 per unit

3. Looping Statements

For loop -> for( int i=1;i<=5;i++ ) {

// statements

for(int j=1;j<=5;j++) {

}

}

For Each => we can discuss later.

While Loop =>

While(condition) {

}

Do while loop:

Do {

}while(condition);

4.Arrays.

Int I = 250;

Int []a = {1,2,3,4,5}; a.length = 5;

Int []a = new int[5];

a[0] = 1;

a[1] = 2;

a[2] = 3;

a[3] = 4;

a[4] = 5;

Arrays tasks:

int arr[] = {1,5,3,2,4};

asc = [1,2,3,4,5];

des = [5,4,3,2,1];

String arr[] = {“Naresh”, “Triveni”,”Veera”, “Sai”};

asc = [“Naresh”, “Sai”, “Triveni”, “Veera”];

desc = [“Veera”, “Triveni”,”Sai”,”Naresh”];

5. Packages

6. Object

If we want to create the memory and save some values we can create object.

7.Methods

Syntax For Method

accessModifier returnType methodName(Parameters) {}

Create Arithmetic class and create methods with return types.

OOPS Concepts:

1.Class

2.Object

3.Inheritence

4.Polymorphism

5.Encapsulation

6.Abstraction

1.Class

Class is Template or Bluprint Or UserDefined datatype => class Human {

String name;

String color;

String gender;

float height;

int weight;

}

2.Object

Object is a instance of class

It is physically exist.

Human h = new Human();

h.name = “naresh”;

h.color = “white”;

h.gender = “m”;

h,height = 5.8;

h.weight = 70;

3.Inheritence

Class Arithmetic {

Addition();

Substraction();

Multiplication();

Division();

}

Class Arithmetic2 extends Arithmetic {

Reminder();

}

Access Modifiers:

1.public: It is acceptable from one package to another package.it contains global access.public we can use at class level, method level and variable level.

2.private: private modifier we can access within the class only. private only methods and variables.

3.protected: protected modifier we can access within the same package and sub class. protected we can use at method level and variable level.

4.default: default modifier within the same package only. default we can use at class level, method level and variable level.

Encapsulation

Encapsulation is nothing but a properties and behaviours we can take as single unit or we can combine both.

26/11/2020

Abstraction:

Abstraction is nothing hiding internal implementation.

Abstraction we can achieve by 2 ways.

1. Interfaces -> 100%
2. Abstract classes -> 0 – 100%

Is it possible to create concrete methods in interface?NO

Is it possible to create concrete methods in abstract classes?Yes

Is it possible to create abstract methods in abstract classes?Yes

Task: Payroll task

Constructors

Constructor is used for to initialize the values into the object we can constructor.

Syntax for constructor:

accessModifier classname(){}

Access Modifiers for Constructor

We can private, protected, default and public as accessmodifiers.

static vs non-static, string

static is a key word, we can use static as a variable, method and block.

Static Variable

* Static variable -> The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.
* The static variable gets memory only once in the class area at the time of class loading.
* Advantages of static variable
* It makes your program **memory efficient** (i.e., it saves memory).

Static Block

* Is used to initialize the static data member.
* It is executed before the main method at the time of classloading.

### Q) Can we execute a program without main() method?

Ans) No, one of the ways was the static block, but it was possible till JDK 1.6. Since JDK 1.7, it is not possible to execute a Java class without the [main method](https://www.javatpoint.com/java-main-method).