**Intro OOPS(Object Oriented Programing)**

When we work on any problem then it is revolve around the real life problem

-in real life we are surrounded by the objects so in E-learing user, batch, course, problem, videos, leactures are the object example if user is a object and it have the give admission is the functionality

So we write the user.giveaddmission()

Every communication will be revolve around the object oriented.entire idea of object programming is revoler around the class and object (what is object is just like a entity like in any calss room their is no. Of people like all the student are like a object, and each student have some set of property. And these property is same for all object so all these property and functionality are coomn for all student and we don’t want to write it every time so their class is coming, class is basically a templet class have only the all property and functionality and its values is once created when the object are created.

-object is an instance of the class (if we call class it means genrel and if we call object it is specific). The communication is in between the objects only then classes for it is for the it ley down for the function and property list.

**CLASSES AND Objects:-**

When you create new file you need to create the new class.

**Package classesandobjetcs;**

**Public class student {**

**String name;**

**Int rollnumber;**

**}**

This is a way to create a class

**Student s1=new student();**

**Student s2=new student();**

new student () means new object is created and Student s1=is the reference of the object. Means s1 and s2 have the sepreat memory which have the reference of the object if s1 print it print the address. Create object is stored in the heap memory and reference of the object se store in the stack memory.

till now the name and rollnumber is the not specify till now so the have their own default values.

And we can specify by the **s1.name=”Anilesh”;**

**S2.rollnumber=20;**

**ACCESS MODIFIERS:-** they are three type of the access modifier access modifier tells us about how can access the set of property and functionality of the class.

-public -they are accessable by every pacakage

-private -restrict property and functionality with in the class.and access only for that class.

-default -they are accessable with in the same package.

If we use class in the same pacakage no need of import statement

If we use class in the different pacakage then we need to import statement.

**GETTER AND SETTER:-**

-So in the above access modifier their is private access modifier in which their is not directly get the property and set so for that we use the function called getter and setter.

- **Package classesandobjetcs;**

**Public class student {**

**Public String name;**

**Private Int rollnumber;**

**Public int getRollNumber(){**

**Return rollnumber;**

**}**

**Public viod setRollNumber(int num){**

**rollNumber=num;**

**}**

**}**

Now for set the rollnumber we have to use the setRollNumber and for get the use getRollNumber

Then rise the one question if we can get and set the property by the getter and setter so why we use private the property we can make it public we can but any one write the inappropriate value. Like roll number =-100 so for that in setter we can make a condition. In the PSVM we write the s1.getRollNumber(100);

**THIS KEYWORD:-**

**This** is reference to the object. Its how bocs when we call the getRollNumber() we don’t call direatly we have to use the s1.getRollNumber means we have object and that this is reference to that object now if we print the this here then it will show the reference of the object.

**This is show no error this show error**

**Public viod setRollNumber(int rollNumber){ Public viod setRollNumber(int rollNumber){**

**This.rollNumber= rollNumber; This.rollNumber= rollNumber;**

**} }**

-this property is use to differ the property of object and local vairiable.

**CONSTRUCTORS:-**

Constructor is like a function but it is called only once.

-Scanner sc=new Scanner (Syetm.in);

-student s1=new student ();

In both the above new Scanner(System.in); and new Student (); both are construcator they are looks like a function call but it differ from construcator because its call only once in the object cycle. The constructor call when the object are created its call only once in the hole object life time in Scanner we need the source but not in the student. And we are not create the **student(){}**

(and this is calledthe default construcator they don’t do any significant change)

Then from where it is created it is created by the java and it is when a class and object created its constructor is called.

We can also create the constructor for us it is do some task not return anything while function do return

Name of constructor is the name or the class same, when we create our own constructor then the default constructor is not longer more their. Then we have to pass the things tand it will set p according to the constructor.

- **Package classesandobjetcs;**

**Public class student {**

**Public String name;**

**Private Int rollnumber;**

**Public Student(String n,int num){**

**Name=n;**

**rollNumber=num;**

**}**

**Public int getRollNumber(){**

**Return rollnumber;**

**}**

**Public viod setRollNumber(int num){**

**rollNumber=num;**

**}**

**}**

Now in the PSVN we create the object by the Student s=new Student(“Anilesh”,50);

If constructor wants the two argument and you want to pass only one use can do it by the help of new constructor created with one argument.

If we give the this keyword then we use the argument and variable same name other wise we have to kept it different.

**FINAL KEYWORD:-**

Something we don’t need to think change every time so fixed once that is secure by the help of the keyword that is final.

**Package classesandobjetcs;**

**Public class student {**

**String name;**

**Int rollnumber;**

**Final double coversionFactor=0.95;**

**}**

It will set only once not change the further not even in the class or out side the class. We can set only once in the life cycle.

In the final we can create only the once not further if we want to then we have to create the reference only by the final int rollNumber then we define in the constructor bocs constructor call only once so that it can create only the once not further it will overwrite.

**Public class Student{**

**Final int rollNumber;**

**Public Student (String name,int rollNumber){**

**This.name=name;**

**This.rollNumber= rollNumber;**

**}**

So we can set the value of the at the time of decleration or we can set at in the constructor so that at the time of constructor call it will set.

**STATIC KEYWORD:-**

Static means this property is shared by all objects but belong to class. it means their is not multiple cope shared by the object it the only writeen in the class and shared by the all object and if print its value its value will be same for all the object.

**Public class Student{**

**Final int rollNumber;**

**Static int Numstudent=0;**

**Public Student (String name,int rollNumber){**

**This.name=name;**

**This.rollNumber= rollNumber;**

**Numstudent++;**

**}**

Excess we don’t need the object we can directly call the class means s1.numstudent is not good way we have to call the student.numstudent is the good way;

And nonstatic property have issue to use excess by the class name like student.name is not right bocs all the student have the sepreate name.

We can use the final static together the it make like a constant bocs a/c to the final its means it is set only once and due to static its belong to class and also final static variable to create in capital letter its the convention and it give to the public bocs its a constant.

**Public final static double CONVERSIONFACTOR =0.95;**

**STATIC FUNCTION:-**

And the static function is not restrict so we can change it from oout side the class so we need to put as private

Private static int numstudents;

And this now can’t be set out side the class but we can get it from out side so we have to set the getter for it. Must be think is that if variable is static and make its getter functoion so we need to use the static also in the function of getter if not it not showing any problem just it then excess by the object

**Public static int getNumStudent(){  
return numStudents;**

**}**

We cant excess the nonstatic variable in the static function bocs the function is call by class but the nonstatic variable is for the objects means

**Public static int getNumStudent(){**

**System.out.print(rollNumber);** (here this is show the error) **return numStudents;**

**}**

Bocs the static is class based not the object based thats why.

You would not need the object the studentuse class we can’t directly excess main via the class.