Classes and Object

Q. What is class?

---

class is combination of state and behavior here state means variable and behavior means function means we can say class is combination variables and functions

or

class is combination of instance variable, class variable, methods , constructor ,instance initializer, static intiailizer, and nested classes.

Instance variable: If we declare variable within a class without static keyword called as instance variable.

class variable: if we declare variable within a class with static keyword called as class variable.

Method: if we define function within class called as method.

Constructor: if we define function same name as class name called as constructor.

Instance initializer: if we define block within class without static keyword called as instance initializer.

static initializer: if we define block within class with static keyword called as static initializer.

Nested classes: if we define class within class called as nested class.

How to declare class in JAVA

If we want to declare class in java we have following syntax

```
Syntax:
class  classname
{
    access specifier    datatype variablename;
    access specifier return type functionname(datatype variablename)
    {
    }
}
```

Now we want to discuss about the above syntax.

class classname: here class is keyword for declare the class and class name means user can give any name to his class.

Example: class Employee;

access specifier data type variablename: we can declare variable in class using this syntax

Example: private int id;

Q. what is access specifier?

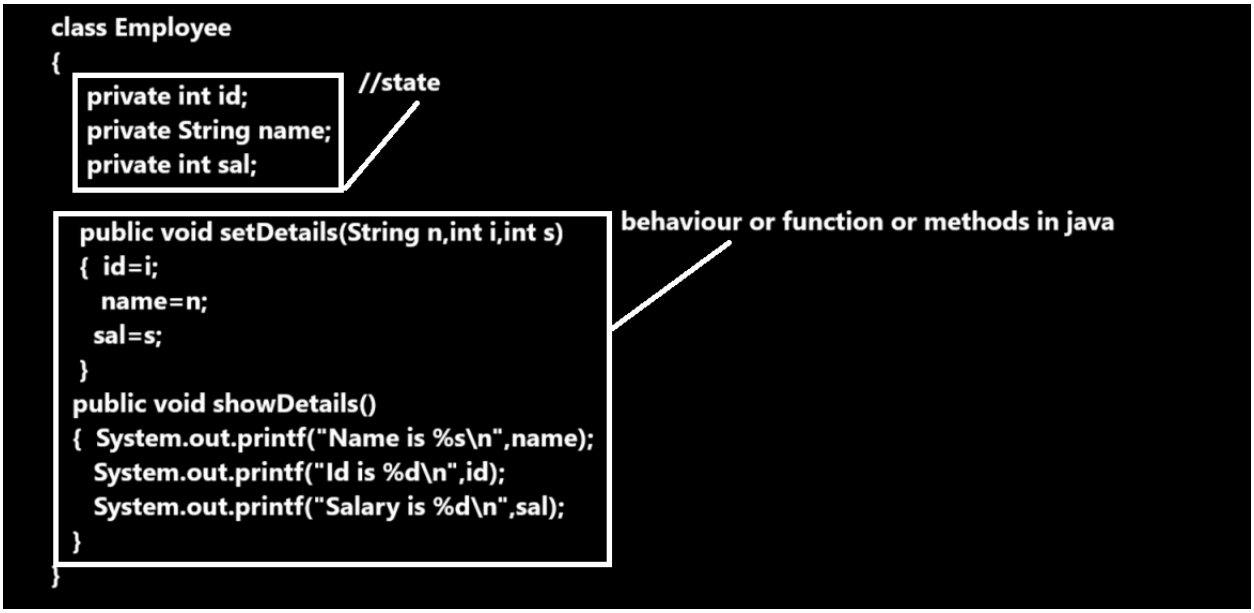
Access specifier is the some keywords which help us to apply restrictions on class and its member.

There are four types of access specifier in JAVA?

---

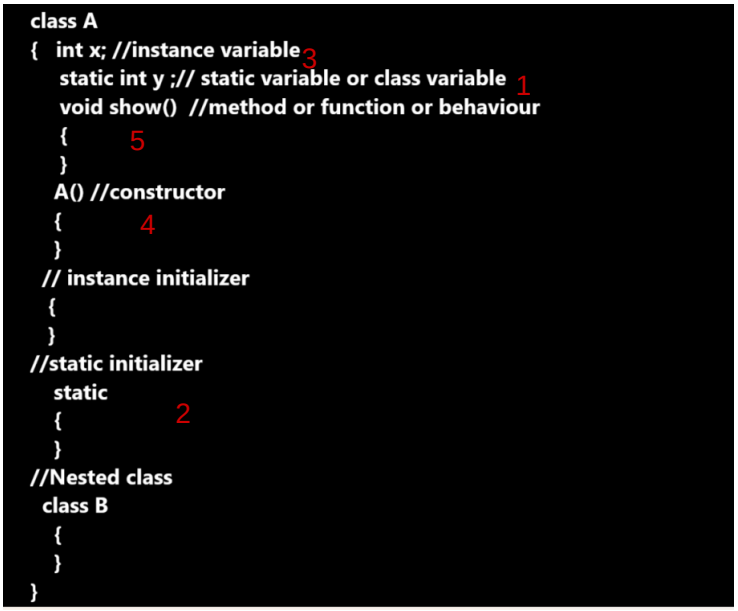
private: private access specifier help us to restrict class member access outside of class means access specifier provide the security to class and help us to implement encapsulation using class.
public: public access specifier means member can access outside of class by using its object within a same package as well as outside of package.
protected: protected access specifier means member can access only in child class not outside of class
default : default access specifier means member can access outside of class but within same package not outside of package.

Example of class as per the above syntax



Now we want to satisfy the in depth definition of class given below.

class is combination of instance variable, class variable, methods , constructor ,instance initializer, static intiailizer, and nested classes.



Interview Question – Date-10/11/2024

Q1. What is class and why use it explain with syntax?

Q2. What is access specifier and explain all access of java with example?

Q3. What is instance variable, class variable, method, constructor, instance initializer, static initializer etc?

Program on Function overloading?

Q1. Write a program to create function name as

void setOperation(int a[],int b[]): this function can accept two integer array and find the union of two array

void setOperation(char a[],char b[]): this function can accept two character array ad find the union of character

Q.2. Write a program to create function name as register

void register(String studName,int age , String schoolName,int fees)

void register(String voterName,int age,String wardName)

Q. What is benefit of class / why use class?

There are three benefits of class

1. Ability to store different of data: if you think about the class it is combination of heterogeneous data and user can store data according to his requirement so this is major reason we can say class is user define data type also.

```
class Employee
{
    private int id;
    private String name;
    private long sal;
}
```

If we think about above code we have class name as Employee and it contain three types of variable like id,name and sal and id has integer type of data , name as string type of data and sal has long type of data

2. Reusability of code: we can declare class only once and we can reuse it more than called as reusability

How we can reuse class more than one time?

We can reuse class more than one time by creating its object.

Q. What is object?

Object is a block of memory where class data store means when we create object of the class then JVM allocate memory for that object and store class data/variable in that memory and after that programmer can use class data or members means we cannot use any class without creating its object.

Bookish definition: object is instance of class or run time entity

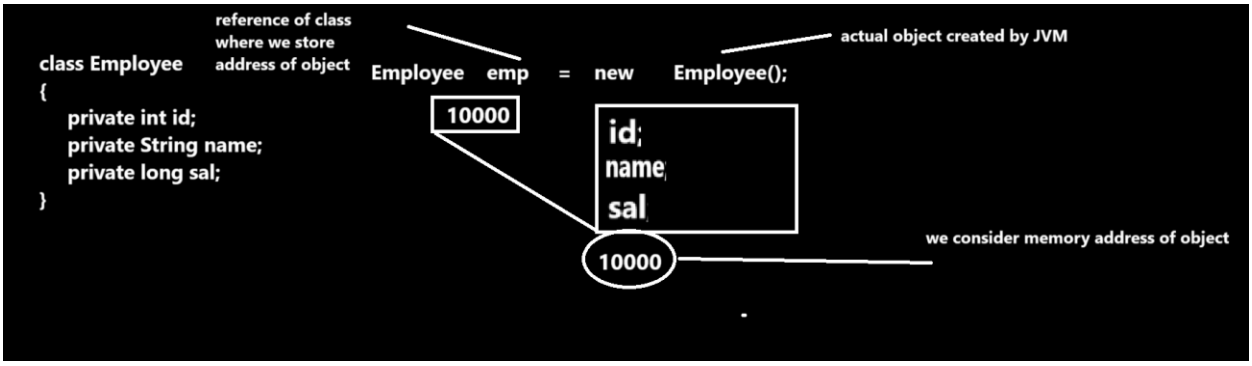
How to create object in JAVA

If we want to create object in java we have following syntax

Syntax: classname ref = new classname();
Example: Employee emp = new Employee();
Note: here emp is reference of Employee class and new Employee() is actual object of Employee class

Q. What is difference between reference and object?

Reference is a variable which hold the address of object and object is a block of memory where class data store shown in following diagram.



Q. Why use reference or what is benefit of reference?

Q. What happen if we not use reference with object?

Note: we will discuss this two question after 3 lectures

Now we want to discuss Steps to use class

---

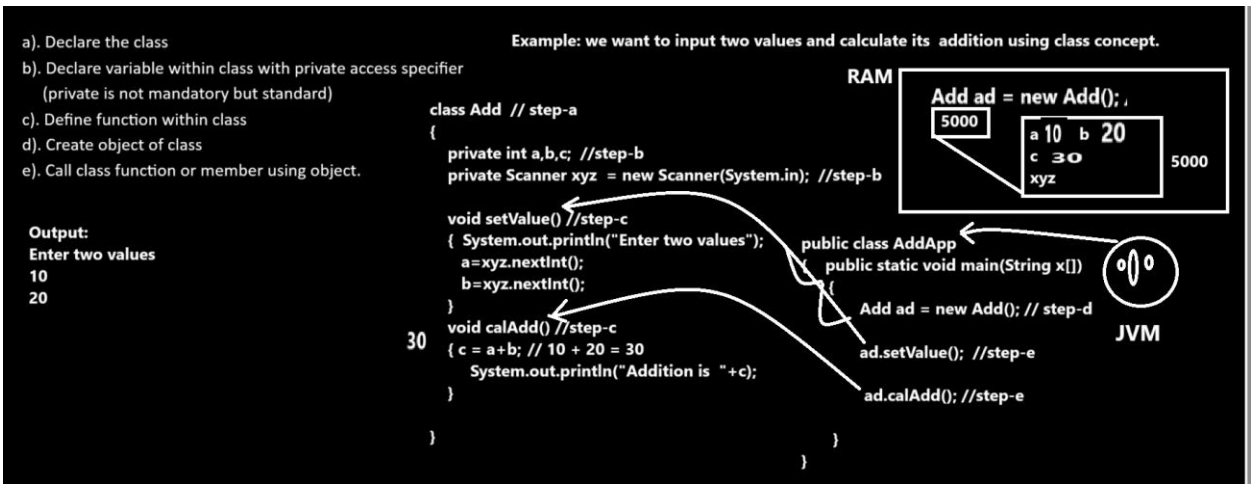
a). Declare the class

b). Declare variable within class with private access specifier
(private is not mandatory but standard)

c). Define function within class

d). Create object of class

e). Call class function or member using object.



Example with source code

```
import java.util.*;
class Add
{
  private int a,b,c;
  private Scanner xyz  = new Scanner(System.in);

  void setValue()
  { System.out.println("Enter two values");
    a=xyz.nextInt();
          b=xyz.nextInt();
  }
  void calAdd()
  {   c=a+b;
    System.out.printf("Addition is %d\n",c);
  }
}
public class AddApp
{  public static void main(String x[])
   {
           Add ad = new Add();
            ad.setValue();
            ad.calAdd();
   }
}
```

Date: 11/11/2024

Interview Question:

Q1. What is object and why use it?

Q2. How we can prove class is user defined data type?

Q3. What is reference and what is object?

Q4. How to create object in java explain with syntax and example?

Example1: WAP to create class name as Square with two functions

void acceptNumber(): this function can accept number from keyboard

void showSquare(): this function can calculate square of number and display it.

Example2: WAP to create class name as Cube with two functions

void acceptNumber(): this function can accept number from keyboard

void showCube(): this function can calculate cube of number and display it.

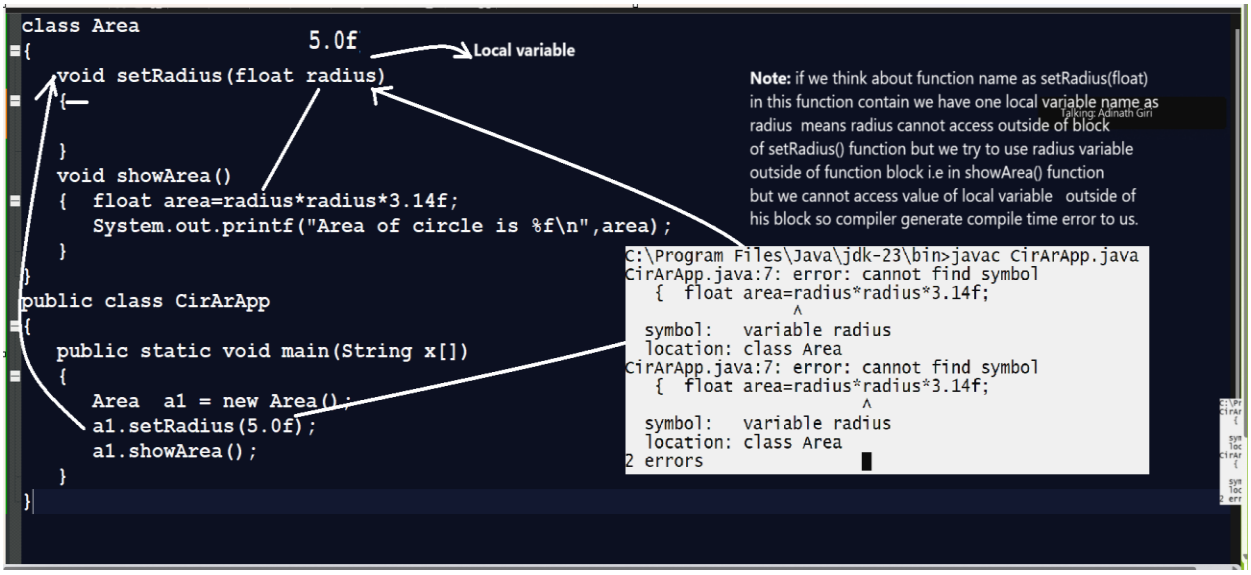
Example3: WAP to create class name as Table with two functions
void setNum(): this function can accept number as parameter
void showTable(): this function can display the table of number

Example4: WAP to create class name as Factorial with two functions
void setValue(): this function can accept the number as parameter
void showFactorial(): this function can calculate factorial and display it.

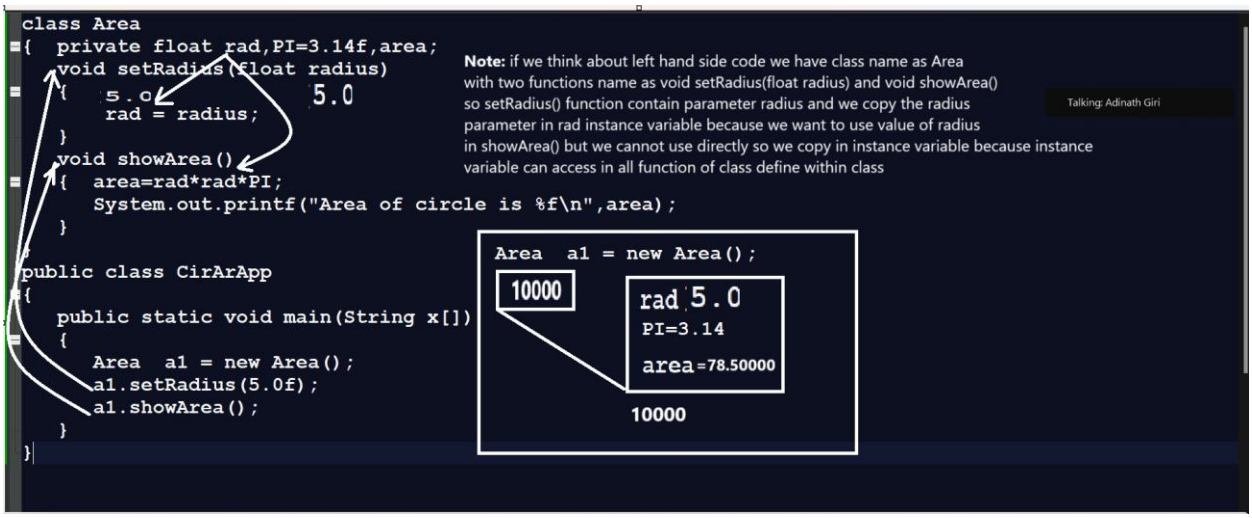
How to pass parameter to call function

If we want to pass parameter to class function then we need to store local variable of function in instance variable for use its value in all class functions.

Example: Create class name as Area with two methods
void setRadius(float radius): this function is used for accept the radius of circle as parameter
void showArea(): this function is used for calculate area of circle and display it.



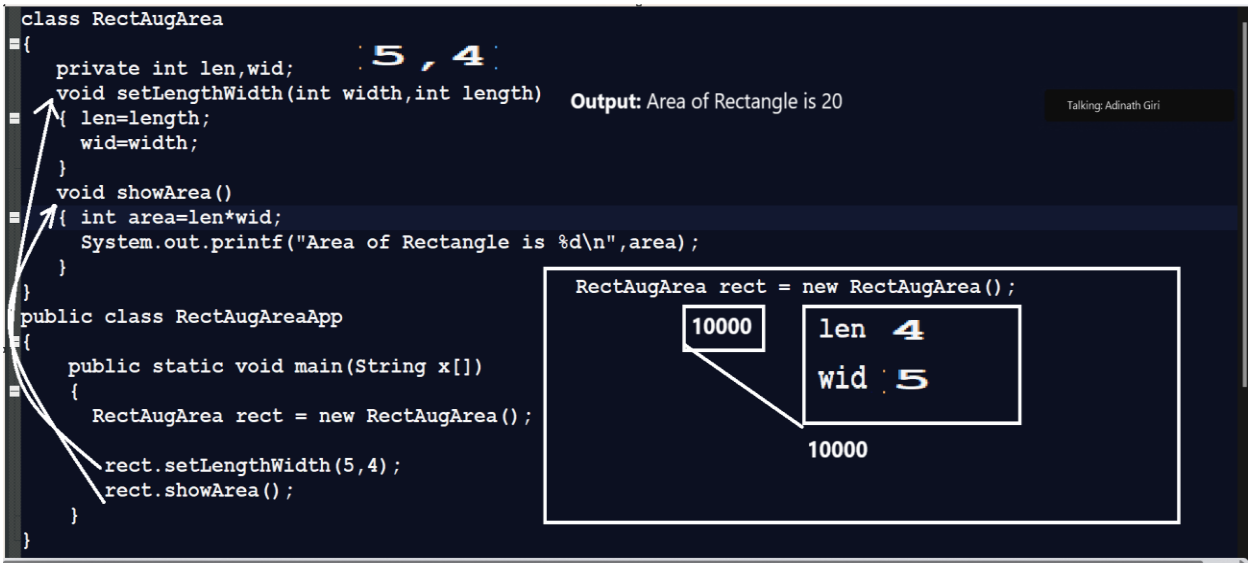
Note: if we want to solve above problem we need to copy value of local variable in Instance variable means we need to declare variable within direct class and copy the value of local variable in instance variable so the benefit instance variable can accessible in all function of that particular class.



Example: WAP to create class name as RectAugArea with two functions

void setLengthWidth(int width,int length): this method accept the two parameter length and width

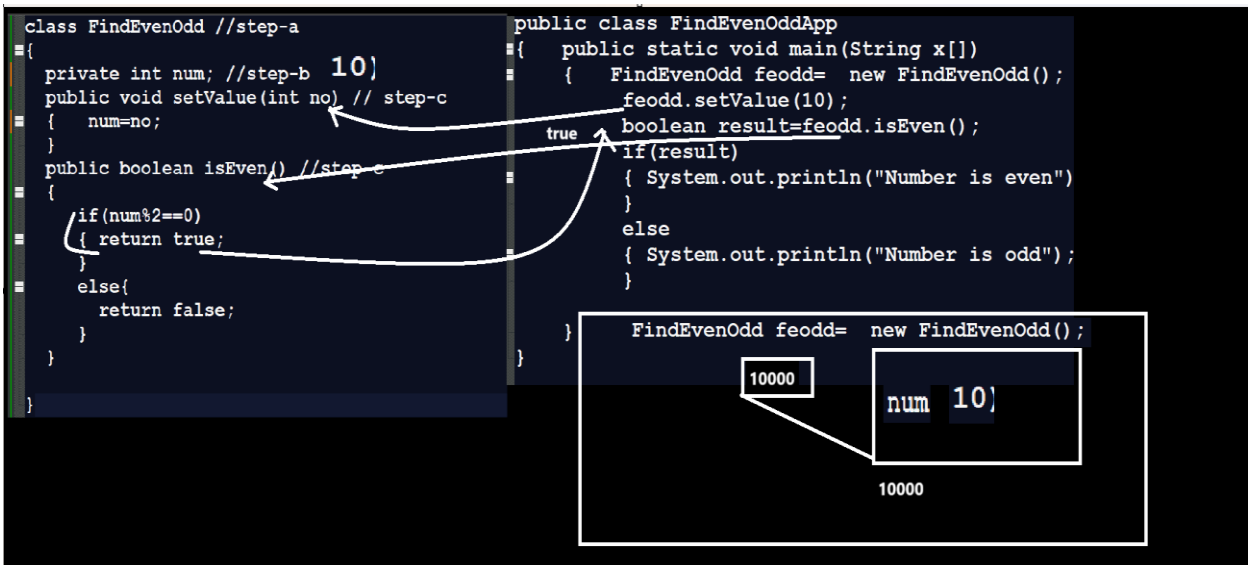
void shwoArea(): this function can calculate area of rectangle and display it.



Q3. WAP to Create class name as FindEvenOdd with two methods

void setValue(int no): this function accept number as parameter

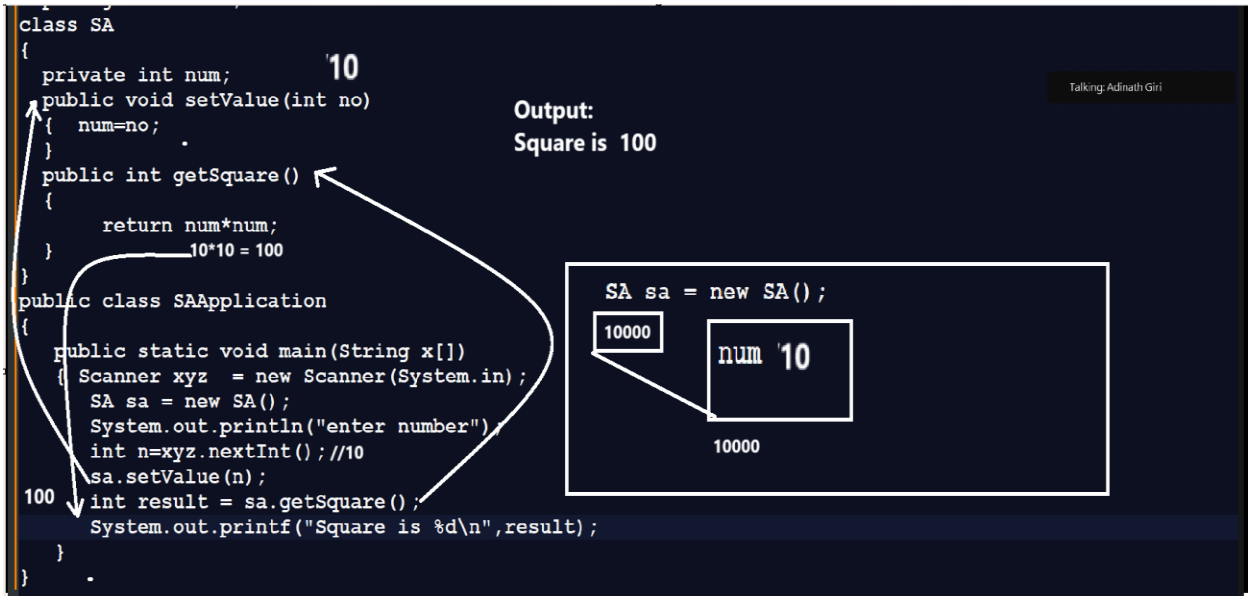
boolean isEven(): this function can check number is even or odd if number is even then return true otherwise return false.



Example: WAP to create class name as Square with two functions

void setNum(int no): this function can accept number as parameter

int getSquare(): this function can calculate the square of number and return it.



Date : 12/11/2024

Programs

Q1. WAP to create class name as Armstrong with two methods

void setNumber(int no): this function can accept number as parameter

boolean isArmstrong(): this function check number is armstrong or not if number is armstrong return true otherwise return false.

Q2. WAP to create class name as DigitCount with two functions

void setValue(int num): this function can accept number as parameter
Int getDigitCount(): this function can count the number digit from function and return it.

Q3. WAP to create class name as Prime with two functions
void setValue(int no): this function can accept number as parameter
boolean isPrime(): this function can check if the number is prime or not if prime return true otherwise return false.

Q4. WAP to create class name as Palimdrome with two functions
void setValue(int no): this function can accept number as parameter
Boolean isPalim(): this function can check if the number is palindrome or not if palindrome returns true otherwise return false.

Q5. WAP to create class name as Fibo with two functions
void setLimit(int limit): this function can set the limit of series
void showFibo(): this function can display the fibonacci series

APTITUDE QUESTIONS

1. The value of a car at the beginning of a year is 10% less than the value of the same car at the beginning of the previous year. If the car is valued at Rs. 1,45,800 on 1st January 2000, what was its value on 1st January 1997?
(1) Rs. 200000 (2) Rs. 250000 (3) Rs. 185000 (4) None of these

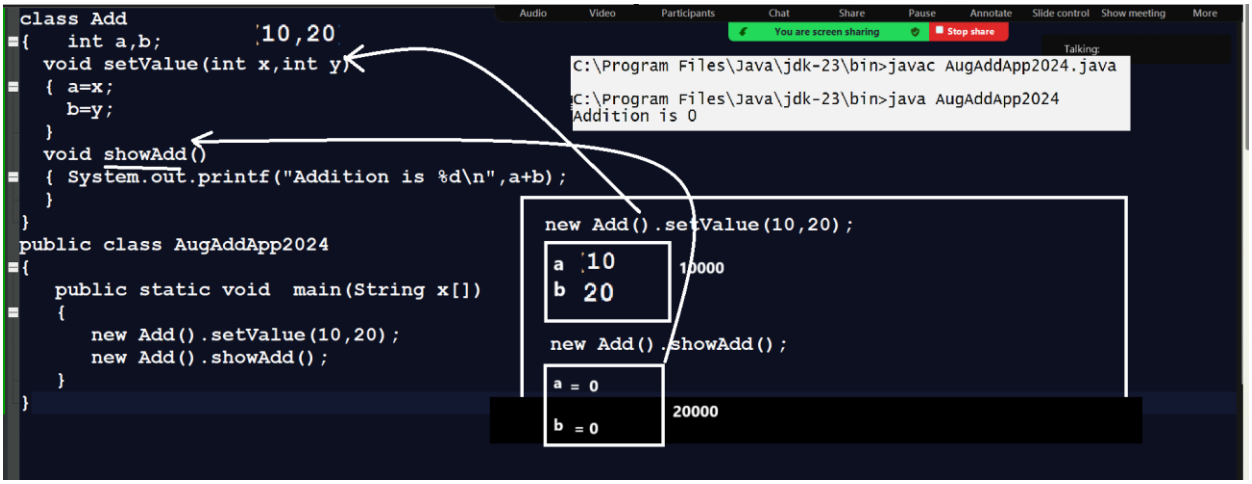
2. In an examination, 42% of the candidates failed in physics, 24% failed in Chemistry, and 14% failed in both subjects. If 72 candidates passed in both, find the total number of candidates in the examination.
(1) 120 (2) 130 (3) 150 (4) 160

3. The number of students who appeared from a school for the Madhyamik Examination in three consecutive years are in the ratio 7: 8: 10 and 75%, 87.5% and 93.75% of the students of respective years were successful. What is the percentage of students who were successful during these three years taken together?
(1) 85.5% (2) 86.5% (3) 87% (4) 88.5%

4. An alloy contains 89% copper; find how much copper must be mixed with a sample of the alloy to get 44 kg of a new metal with 90% copper in it.
(1) 2 kg (2) 3 kg (3) 2.5 kg (4) 4 kg

5. If the numerator of a fraction is increased by 150% and the denominator is increased by 300% the resultant fraction is 5/18. What is the original fraction?
(1) 4/9 (2) 8/9 (3) 6/9 (4)None of these

Q1. What happen if we not use reference with object?



If we think about above code we not use reference with object so the result is we get answer is 0 and we except answer is 30 but if we think about statement new Add().setValue(10,20) the meaning of this statement we have object in memory whose address is 10000 and we call setValue() function using this object and pass two parameter in it 10 and 20 means here 10 and 20 stored in 10000 location variable a=10 and b=20 and again if we think about statement new Add().showAdd() the meaning of this statement is we create new object in memory whose address is 20000 and if we think about 20000 location then there is also two new variables allocated by JVM in object name as a and b with default value 0 so we have statement in showAdd() System.out.printf("Addition is %d\n",a+b) so we get ansower is Addition is 0
Means we can say when we not use the reference with object the JVM create new object in memory every time means we have new block in every time in memory means we can say we cannot use same object more than one time in application.
So if we want to same object more than one time then we need to use reference with object in program

Note: When we not use reference with object then programmatically the name of object is known as anonymous object.

Q. What is anonymous object?

Anonymous object means object without reference called as anonymous object.

Q. Why use anonymous or when use anonymous object?

If we want to use object only in application or when we call single member from class using that object and not required in future so better creating object with reference is not good approach so best you can create object without reference known as anonymous object

Q. what is new keyword?

When we use new keyword then JVM create new memory block every in ram or In JVM heap space and JVM heap space is part of RAM

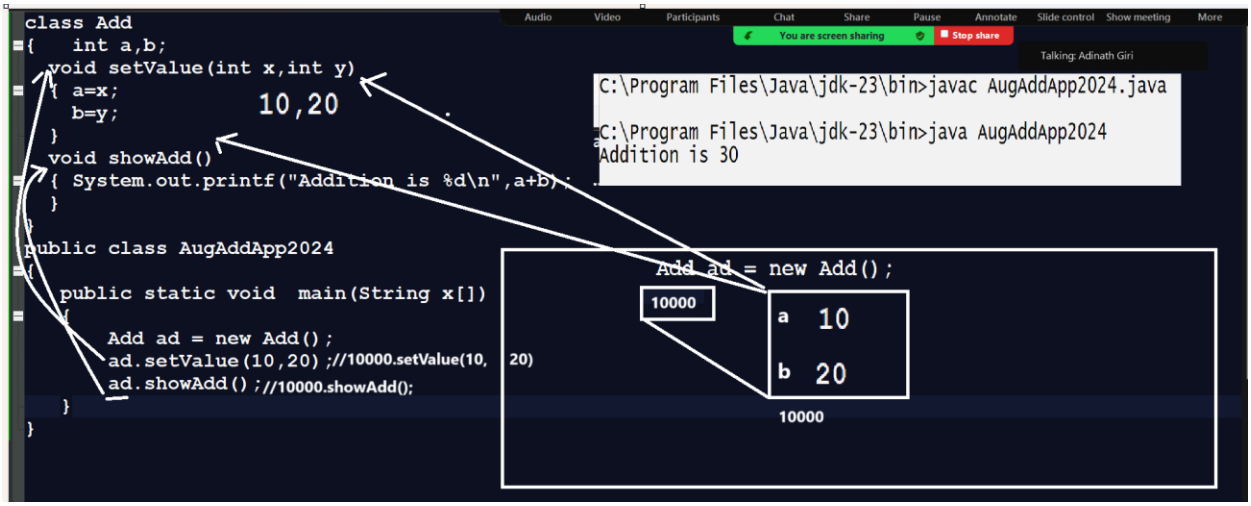
When we use new keyword with object then JVM create new object every time of that class whose name associated with new keyword or whose constructor associated with new keyword

When we use the new keyword with array then JVM create new array every time in memory of specified data type according to requirement of user

Q2. Why use reference with object or what is benefit of reference?

If we want to use same object more than one time then we can use reference with object.

Example:



Note: if we think about above code we have statement Add ad = new Add() means here we create object of Add class whose address we consider 10000 and we store this address in ad reference and we have statement ad.setValue(10,20) means 10000.setValue(10,20) means we call object using address 10000 and we pass 10 and 20 to function and using function we store these values in instance variable a and b means we can say after ad.setValue(10,20) we have 10 and 20 object and again we have statement ad.showAdd() means we not create object new

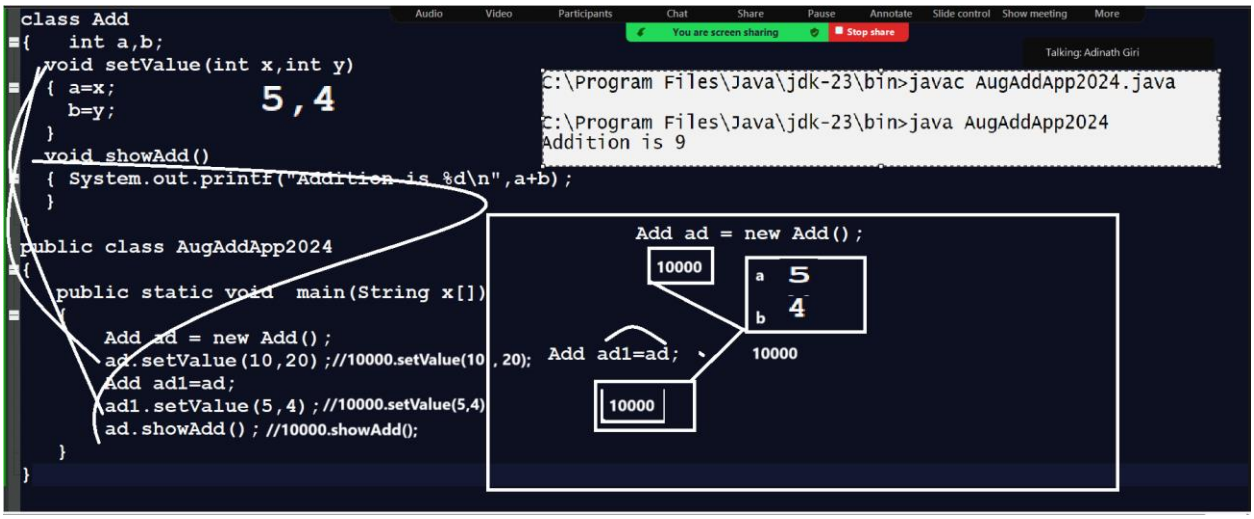
object of class we use previous object of class whose address stored in ad reference i.e 10000.showAdd() means we reuse object again with showAdd() function whose address is 10000 i.e reference is 10000 and if we think about object whose address is 10000 in that object value of a and b is 10 and 20 so when we use statement System.out.printf("Addition is %d\n",a+b) means 10+20 so we get answer Addition is 30
According to this example we can say if we want to same object more than one time then we can use reference with object.

Q3. Can we use the more than one reference with single object and what happen if we use it?

Yes we can use more than one reference with same object and if we give the more than one reference with single object and if we customize object or change values or change state of object by using any reference then object previous state may be loss or previous values may be loss and for resolve this problem we have object cloning concept in JAVA

Note: we will discuss object clone concept after 2 months in Object class concept

Now using a following code example we have discuss meaning of above statement

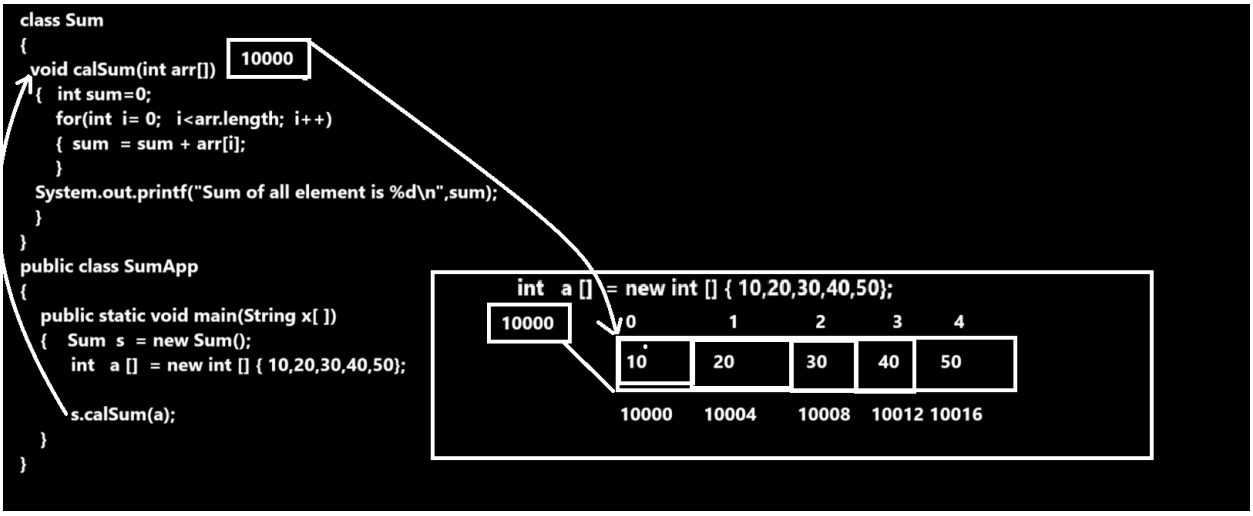


Note: if we think about above code we have statement Add ad = new Add() means we create Add class object in memory whose address we consider as 10000 and we store in ad reference and we have statement ad.setValue(10,20) means 10000.setValue(10,20) means 10 and 20 get stored in object whose address is 10000 means a=10 and b=20 on 10000 location and again we have statement Add ad1=ad; means we copy the address of ad in ad1 means ad1 is also 10000 means ad and ad1 both references points to same object so we have one statement ad1.setValue(5,4) means 10000.setValue(5,4) so 5 and 4 get stored on 10000 so after this statement a=5 and b=4 on 10000 means we can say 5 get override on 10 and 4 get override on 20 means we lose the previous values of object whose address is 10000 because we change object by another reference whose name is ad1 so after that we have statement ad.showAdd()

Means 10000.showAdd() and in showAdd() we have System.out.printf(“Addition is %d\n”,a+b) means 5+4 so we get answer Addition is 9

How to pass array as parameter to class function

If we want to pass array as parameter to class function so we have to pass base address in class Function

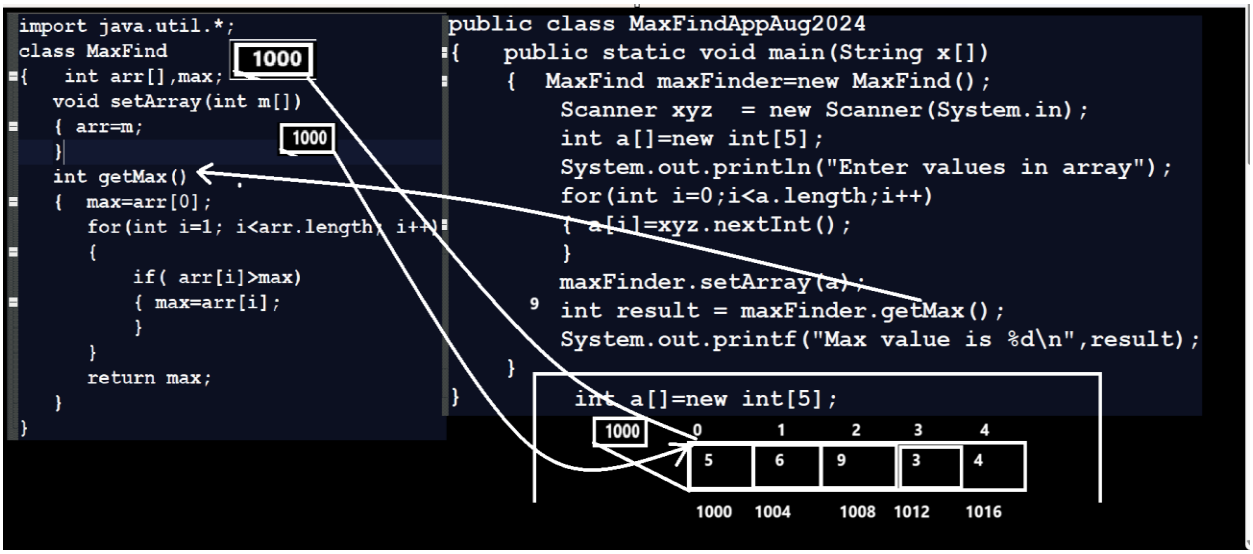


Example with source code

```
class Sum
{
  void calSum(int arr[])
  { System.out.println("Base address of arr "+System.identityHashCode(arr));
     int s=0;

          for(int i=0; i<arr.length; i++)
          {
            s= s+ arr[i];
          }
          System.out.println("Sum of all value is "+s);
  }
}
public class AugSumApp2024
{
   public static void main(String x[])
         {
           Sum s = new Sum();
```

```
        int a[]=new int[]{10,20,30,40,50};
            System.out.println("Base address of arr "+System.identityHashCode(a));

            s.calSum(a);
        }
}
```

Example: WAP to create class name as FindMax with two methods
void setArray(int arr[]): this method can accept array as parameter
int getMax(): this method can find the max value from array and return it.



Example: WAP to create class name as Search with two methods
void setArray(int arr[]): this function can accept array as parameter
int search(int key): this function can accept key as parameter and search element and return its index otherwise return -1.

Example with source code

```
import java.util.*;
class Search
{  int arr[],index=-1;
   void setArray(int a[])
   {  arr=a;
   }
   int search(int key)
   {   for(int i=0;i<arr.length;i++)
```

```
        {
                        if(arr[i] == key)
                         { index=i;
                         }
        }
        return index;
  }
}
public class SearchArrAug2024
{
  public static void main(String x[])
  {       Search s = new Search();
          Scanner xyz  = new Scanner(System.in);
          int a[]=new int[5];
          System.out.println("Enter values in array");
          for(int i=0;i<a.length;i++)
          {   a[i]=xyz.nextInt();
          }
           s.setArray(a);
                System.out.println("Enter search value");
                int svalue=xyz.nextInt();
          int index=s.search(svalue);
          if(index!=-1)
          {System.out.println("Data found  "+a[index]);
          }
          else
          {System.out.println ("Data not found");
          }
  }
}
```

Output

```
C:\Program Files\Java\jdk-23\bin>javac SearchArrAug2024.java

C:\Program Files\Java\jdk-23\bin>java SearchArrAug2024
Enter values in array
10
20
30
40
50
Enter search value
30
Data found  30
```

Method with variable arguments

Method with variable argument means we can pass infinite parameter to the function
Means user can pass values according to his choice from function calling point.

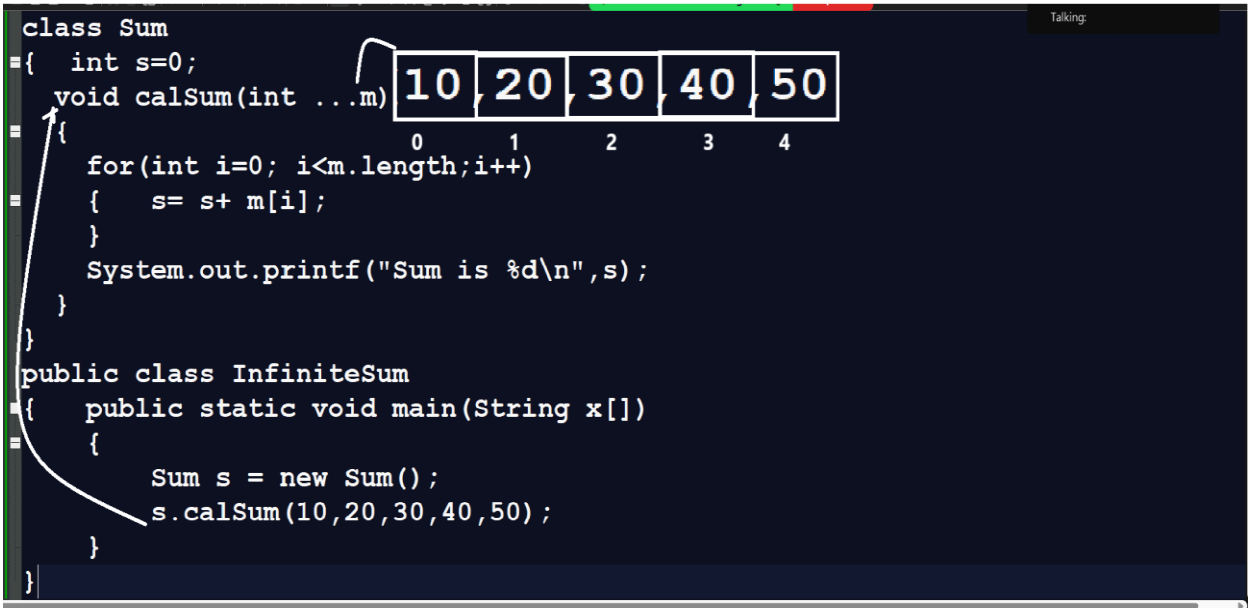
Syntax:

```
return type  functionname(datatype ...variablename)
{

}
```

infinite arguments

Example related with variable argument

We want to design one function accept the infinite number of integers and can calculate its sum.




```
class Sum
{  int s=0;
   void calSum(int ...m)
   {
      for(int i=0; i<m.length;i++)
      {    s= s+ m[i];
      }
      System.out.printf("Sum is %d\n",s);
   }
}
public class InfiniteSum
{   public static void main(String x[])
    {
        Sum s = new Sum();
        s.calSum(10,20,30,40,50);
    }
}
```

Important points related with variable arguments:

a) Here … indicate the variable argument or we can pass infinite parameter to function

b) … indicate it is run time array mean internally all parameter pass from calling point accept as array in function definition

c) … must be pass left hand side of variable and right hand side of data type.

Example: void calSum(int m...) : it will generate compile time error to us

d) We cannot pass more than one variable arguments in single function

```
class Sum
{  int s=0;
   void calSum(int ...m,float ...y)
   {
     for(int i=0; i<m.length;i++)
     {    s= s+ m[i];
     }
     System.out.printf("Sum is %d\n",s);
   }
}
public class InfiniteSum
{   public static void main(String x[])
    {
        Sum s = new Sum();
        s.calSum(10,20,30,40,50,5.4f,2.5f,5.6f);
    }
}
```

```
C:\Program Files\Java\jdk-23\bin>javac InfiniteSum.java
InfiniteSum.java:3: error: varargs parameter must be the last parameter
  void calSum(int ...m,float ...y)
                      ^
1 error
```

Note: if we think about left hand side code we get compile time error because we pass two variable arguments in single function and it is opposite policy of variable arguments

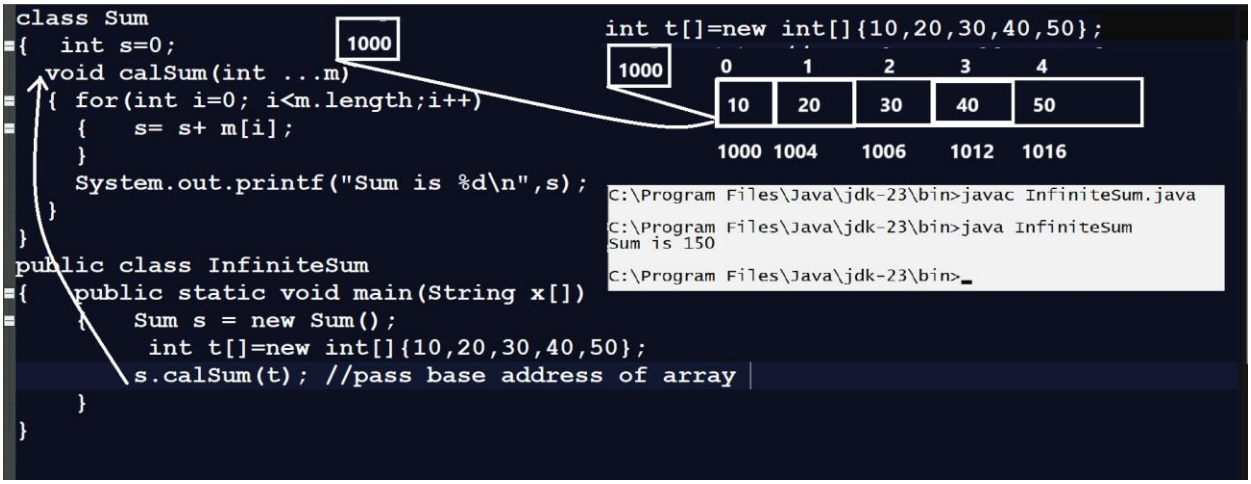
so we get compile time error

e) If we have some simple parameter in function and some variable arguments in function then Variable argument must be last parameter.

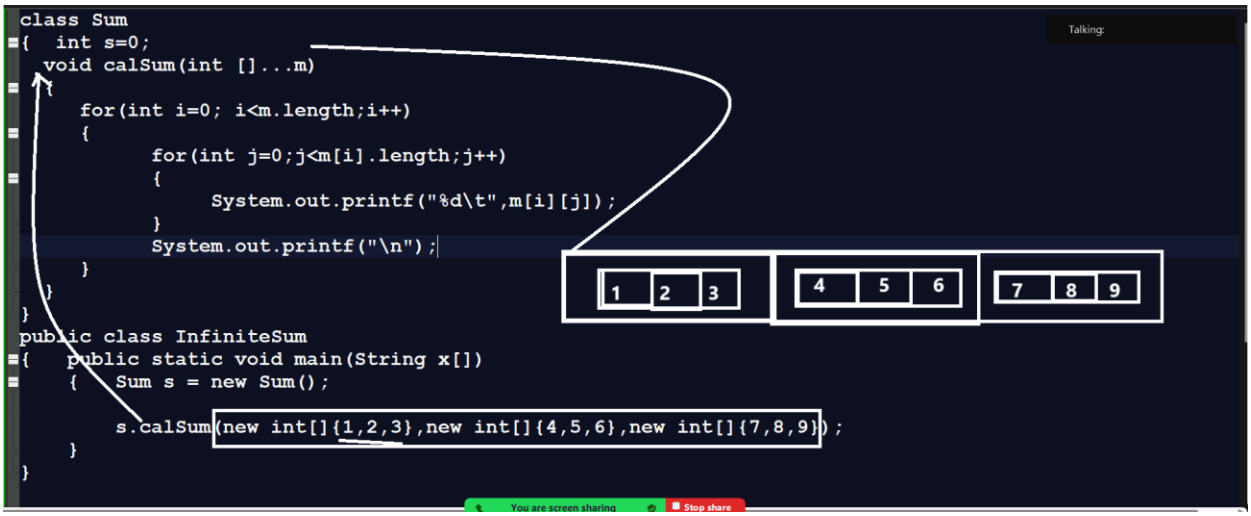
```
class Sum
{  int s=0;
   void calSum(String name,int ...m)
   { System.out.println("Name is "+name);
     for(int i=0; i<m.length;i++)
     {    s= s+ m[i];
     }
     System.out.printf("Sum is %d\n",s);
   }
}
public class InfiniteSum
{   public static void main(String x[])
    {
        Sum s = new Sum();
        s.calSum("ABC",10,20,30,40,50);
    }
}
```

```
C:\Program Files\Java\jdk-23\bin>javac InfiniteSum.java

C:\Program Files\Java\jdk-23\bin>java InfiniteSum
Name is ABC
Sum is 150
```

You can pass array as parameter if we have variable argument shown in following code



Demo Example



Note: if we think about above code the limitation is we cannot pass different type of data in function using variable arguments

So if we want to solve this problem then we can use POJO class concept for passing different type of data using variable argument concept.

Note: Before that we should have to know what POJO class is

POJO: POJO stands for Plain Old Java Objects

Basically POJO class with setter and getter method and the purpose of POJO class is to store data in object and retrieve data from object means we can say POJO works as container means in POJO class we can store data pass to function

Important points of POJO class

---

a) Declare variable as private

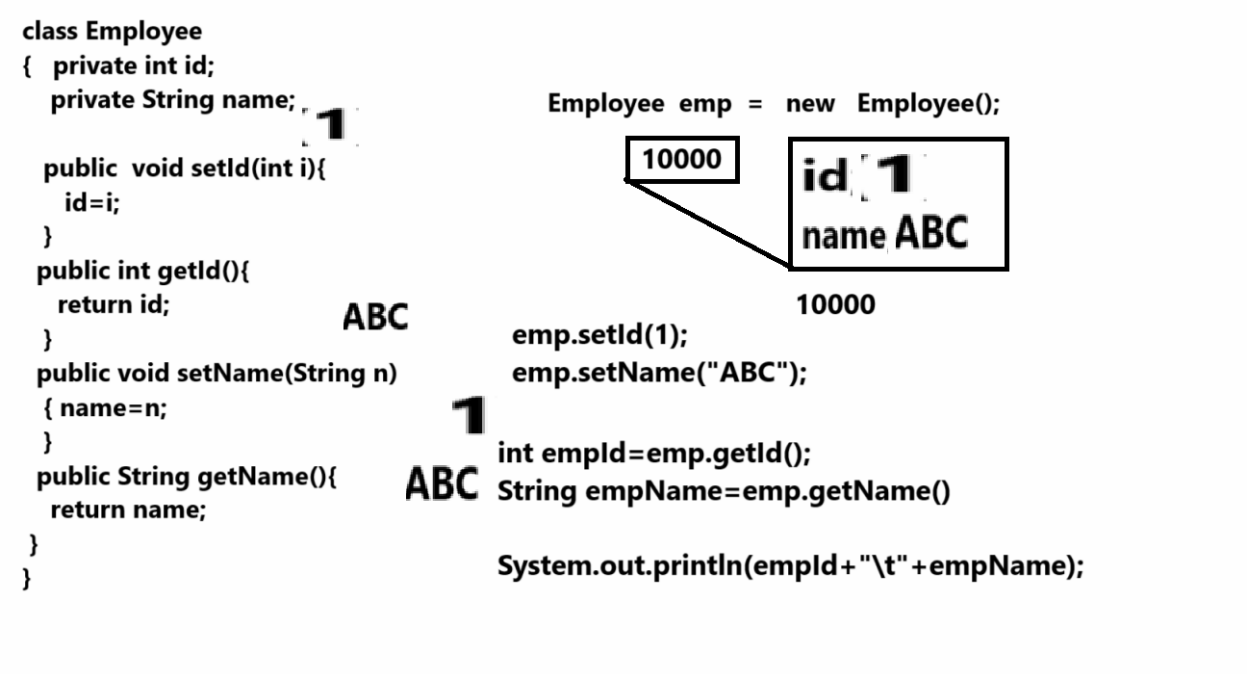
b) When we define method then define by set and get with variable name

c) Normally POJO pass as parameter to function when we want to pass parameter to Function of different type and the parameter list is more than three.

d) It Act as container means we can store different type of data in POJO class object

e) Using setter method we can store data in POJO object and using getter method we can retrieve data from POJO class object.

f) POJO class is perfect implementation of encapsulation in JAVA.



Example with source code

```
class Employee
{
  private int id;
  private String name;

  public void setId(int i)
  { id=i;
  }
  public int getId(){
    return id;
  }
```

```
  public void setName(String n)
  { name=n;
  }
  public String getName()
  { return name;
  }
}
public class EmpPOJOAPP
{
   public static void main(String x[])
          {
             Employee emp = new Employee();
                   emp.setId(1);
                   emp.setName("ABC"); //set data in object or store data in object.
                   int empId=emp.getId();  //fetch data from object
                   String name=emp.getName();
                   System.out.println("Employee Id is  "+empId);
                   System.out.println("Employee Name is "+name);
          }
}
```

Example: WAP to create POJO class name as Player with field id,name and run and store data in POJO class object and display it.

```
class Player
{  private int id;
   private String name;
   private int run;

  public void setId(int i)
   { id=i;
   }
  public int getId(){
     return id;
   }
  public void setName(String n)
   { name=n;
   }
  public String getName(){
     return name;
  }
  public void setRun(int r)
   { run =r;
   }
  public int getRun()
   { return run;
   }
}
```

```
Player p      = new Player();

10000         id     =1
              name   =ABC
              run    = 1000

10000
p.setId(1);
p.setName("ABC");
p.setRun(1000);

System.out.println(p.getId()+"\t"+p.getName()+"\t"+p.getRun());
```

Example:

```
class Player
{
  private int id;
  private String name;
  private int run;

  public void setId(int i)
  { id=i;
  }
  public int getId()
  { return id;
  }
  public void setName(String n)
  { name=n;
  }
  public String getName()
  { return name;
  }
  public void setRun(int r)
  { run=r;
  }
  public int getRun()
  { return run;
  }

 }
public class PlayerApp
{
   public static void main(String x[])
          {
             Player p1= new Player();
                   p1.setId(1);
                   p1.setName("RAM");
                   p1.setRun(1000);

                   System.out.println(p1.getId()+"\t"+p1.getName()+"\t"+p1.getRun());
          }
```

}

Q. Why use POJO class?

Before understanding the need of POJO class we want to discuss one example.

Example: we have class name As Company which contains two functions.
void addNewEmployee(String n,int i,String add,String dob,String prevComp,int sal):
this function can use for add new employee detail
void showDetail(): this function can show the detail of employee

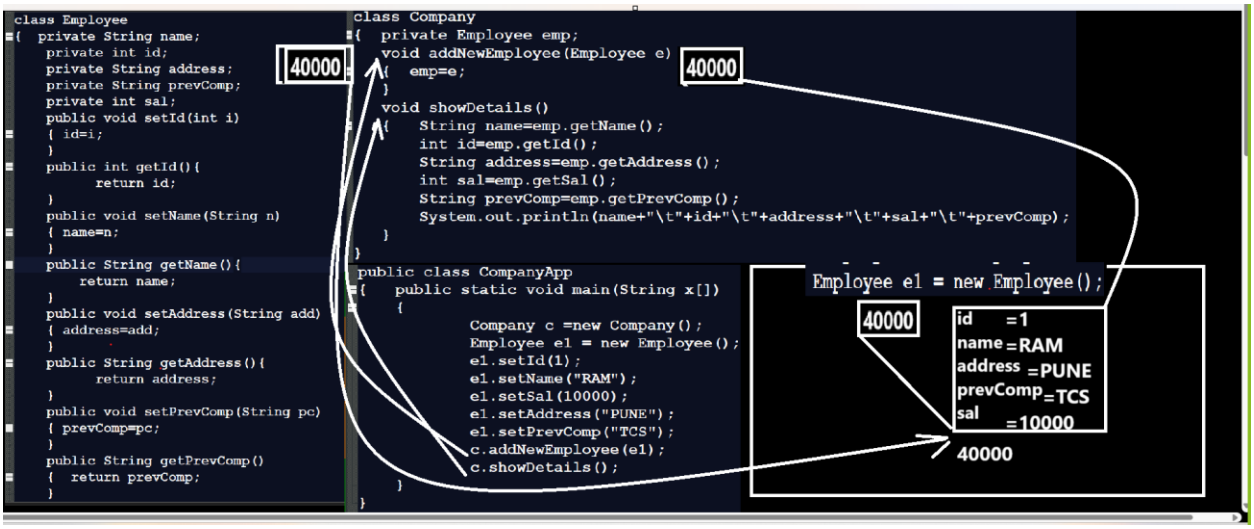
```
class Company
{   private String name;
    private int id;
          private String address;
          private String prevComp;
          private int sal;
  void addNewEmployee(String n,int i,String add,String pcomp,int s)
  {  name=n;
     id=i;
           address=add;
           prevComp=pcomp;
           sal=s;
  }
  void showDetails()
  {  System.out.println("Name is "+name);
           System.out.println("Id is  "+id);
           System.out.println("Address is "+address);
           System.out.println("Previous Company "+prevComp);
           System.out.println("Salary is  "+sal);
  }
}
public class CompanyApp
{   public static void main(String x[])
          {
                        Company c =new Company();
                        c.addNewEmployee("ABC",1,"PUNE","",100000);
                        c.showDetails();
```

}
}

Note: if we think about above code we have function name as
void addNewEmployee() which contain 5 parameters and we call the function then we have some limitations

a) Need to remember sequence of parameter with its type
b) Must be pass all parameter from function calling point to function definition
c) Must be match sequence and data type of parameter

Avoiding this problem java suggest use the POJO class means java create one standard when we have function which three parameter or more than three parameter of different type then passing single sinlge parameter is not good approach better way can use POJO class or model class.



Example with source code

class Employee
{ private String name;
private int id;
private String address;
private String prevComp;
private int sal;
public void setId(int i)
{ id=i;
}

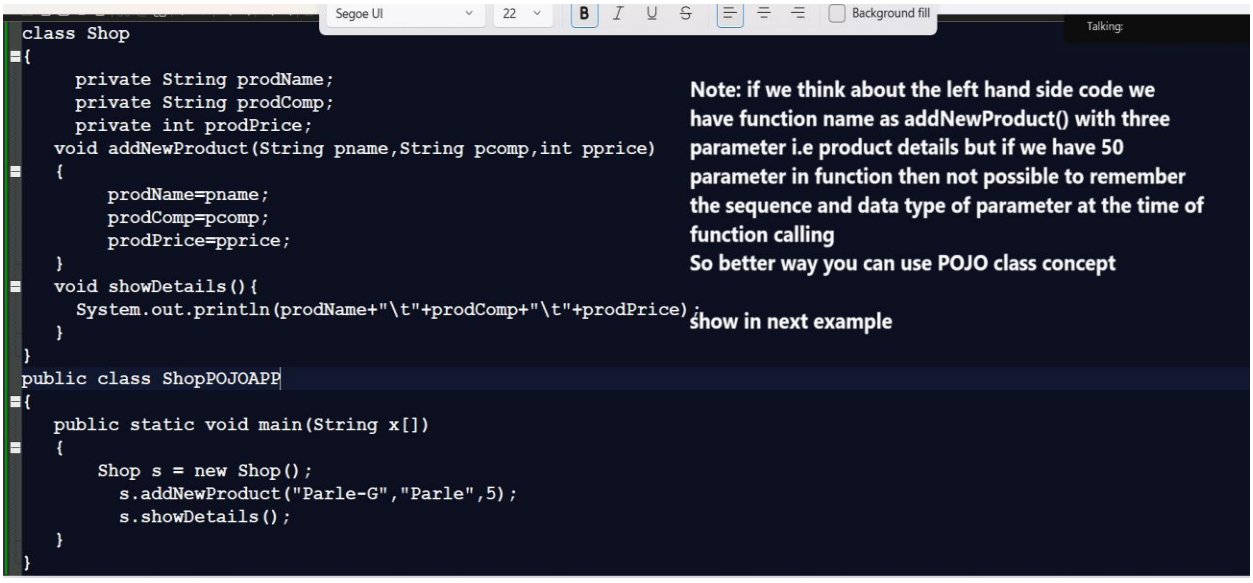
```
        public int getId(){
                return id;
        }
        public void setName(String n)
        { name=n;
        }
        public String getName(){
                return name;
        }
        public void setAddress(String add)
        { address=add;
        }
        public String getAddress(){
                return address;
        }
        public void setPrevComp(String pc)
        { prevComp=pc;
        }
        public String getPrevComp()
        {  return prevComp;
        }
        public void setSal(int s)
        { sal=s;
        }
        public int getSal(){
                return sal;
        }
}
class Company
{  private Employee emp;
  void addNewEmployee(Employee e)
  {  emp=e;
  }
  void showDetails()
  {   String name=emp.getName();
          int id=emp.getId();
           String address=emp.getAddress();
           int sal=emp.getSal();
```

```
            String prevComp=emp.getPrevComp();
            System.out.println(name+"\t"+id+"\t"+address+"\t"+sal+"\t"+prevComp);
  }
}
public class CompanyApp
{       public static void main(String x[])
        {               Company c =new Company();
                        Employee e1 = new Employee();
                        e1.setId(1);
                        e1.setName("RAM");
                        e1.setSal(10000);
                        e1.setAddress("PUNE");
                        e1.setPrevComp("TCS");
                        c.addNewEmployee(e1);
                        c.showDetails();
        }
}
```

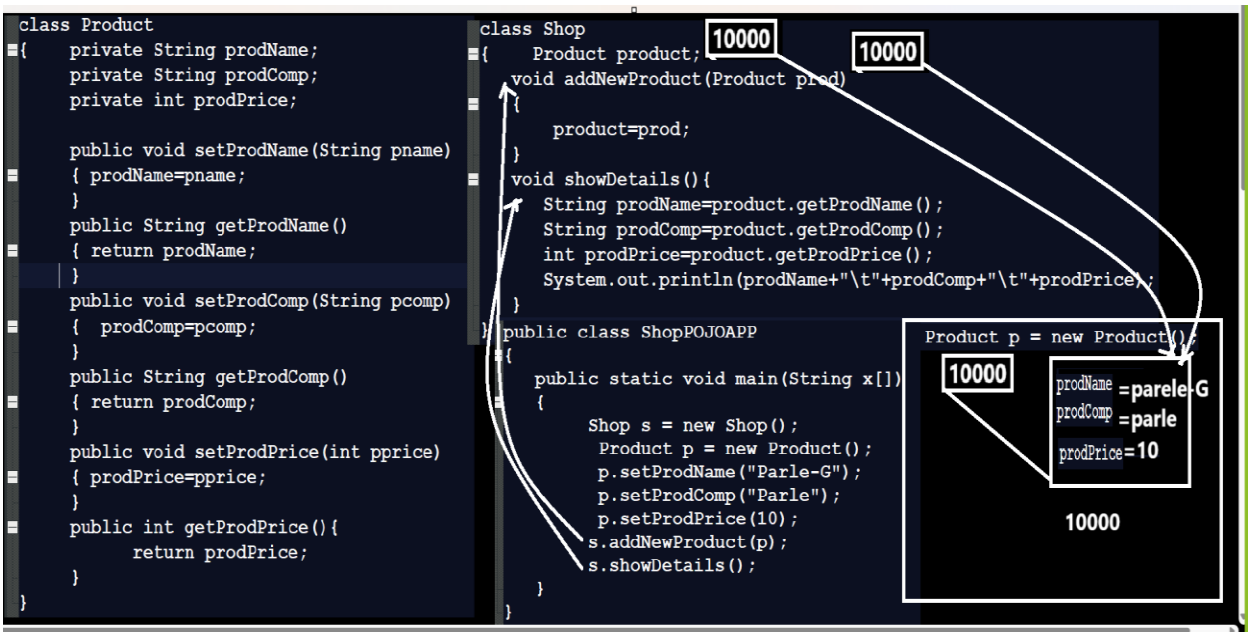
Example: WAP to create class name as Shop with two methods

void addNewProduct(String prodName,String prodComp,int price): this method can accept product info as parameter

void showProduct(): this method can show the product details



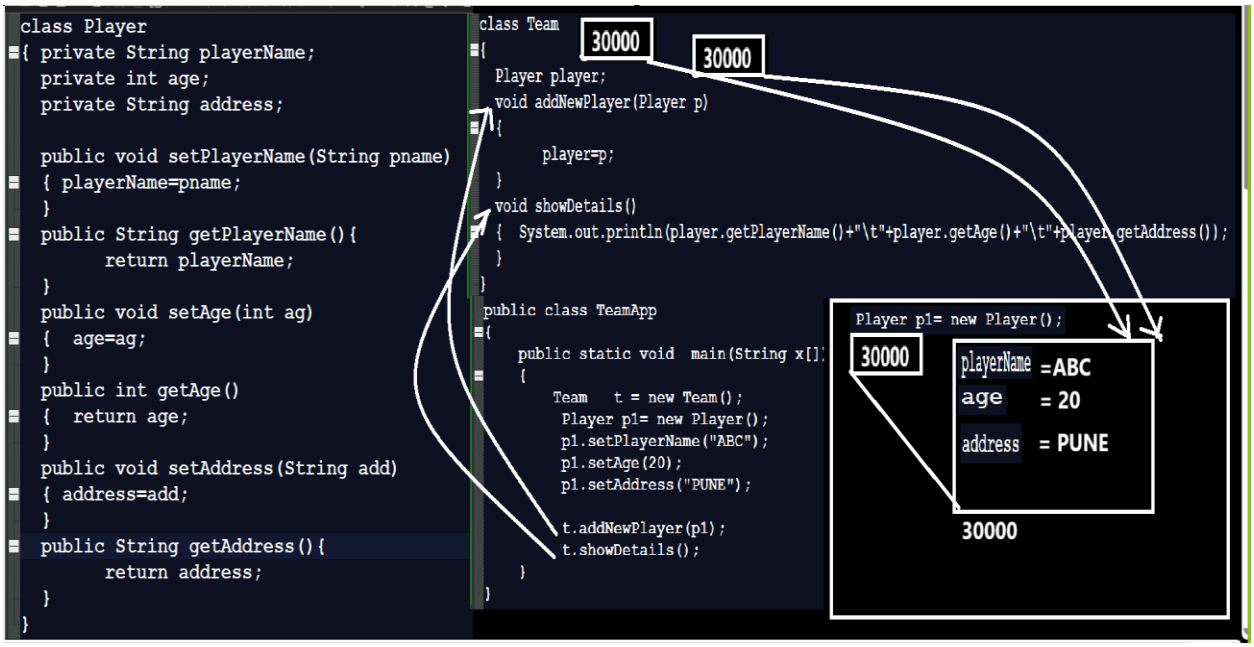
Example with POJO class



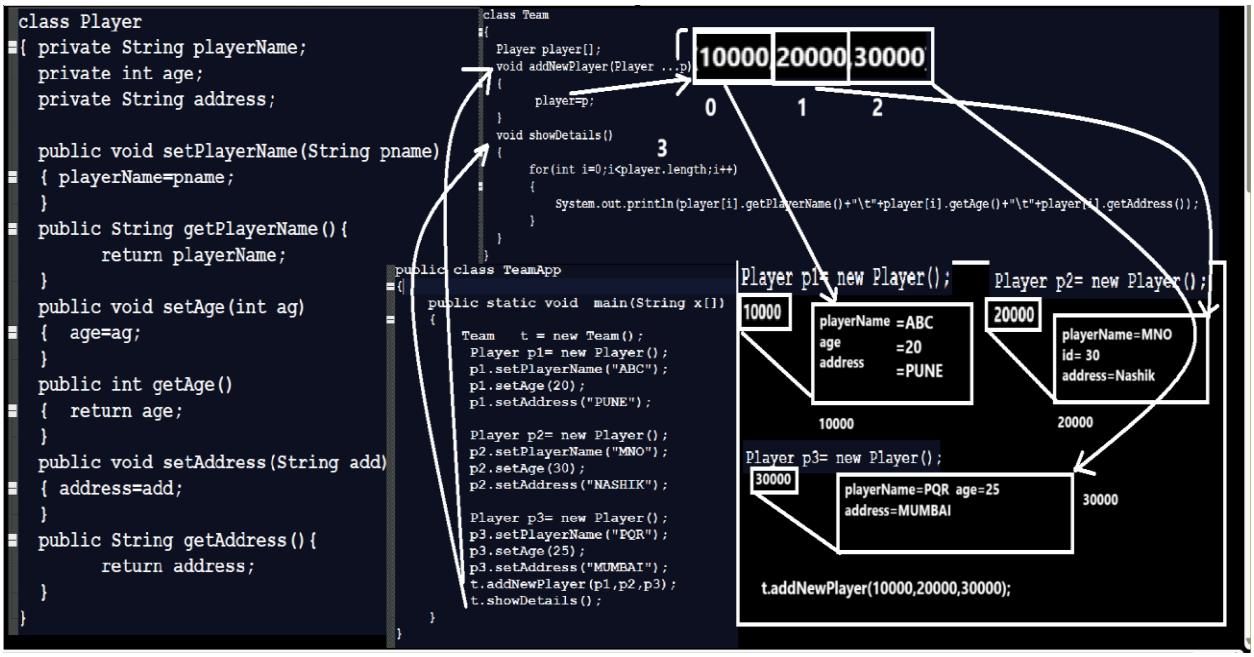
Example: WAP to create class name as Team with two functions

void addNewMember(String name,int age,String address): this function can accept player details

void showDetail(): this function can show the details of the player.



Variable argument with POJO



Example with source code

```
class Admission
{
  private int id;
  private String name;
  private int fees;
  public void setId(int i)
  { id=i;
  }
  public int getId(){
    return id;
  }
  public void setName(String n)
  { name=n;
  }
  public String getName(){
   return name;
  }
  public void setFees(int f)
  { fees=f;
  }
  public int getFees(){
    return fees;
```

```
  }
}
class Batch
{  Admission []adm;
  void setBatchAdmission(Admission ...ad)
  { adm=ad;
  }
  void show()
  {
    for(int i=0; i<adm.length;i++)
          {
           System.out.println(adm[i].getId()+"\t"+adm[i].getName()+"\t"+adm[i].getFees());
          }
  }
}
public class BatchApp
{  public static  void main(String x[])
  {
     Batch b = new Batch();
                  Admission a1 = new Admission();
                  a1.setId(1);
                  a1.setName("ABC");
                  a1.setFees(5000);

                  Admission a2 = new Admission();
                  a2.setId(2);
                  a2.setName("PQR");
                  a2.setFees(10000);

                  Admission a3=new Admission();
                  a3.setId(3);
                  a3.setName("XYZ");
                  a3.setFees(20000);

                  b.setBatchAdmission(a1,a2,a3);
                  b.show();
  }
}
```

Array of object concept

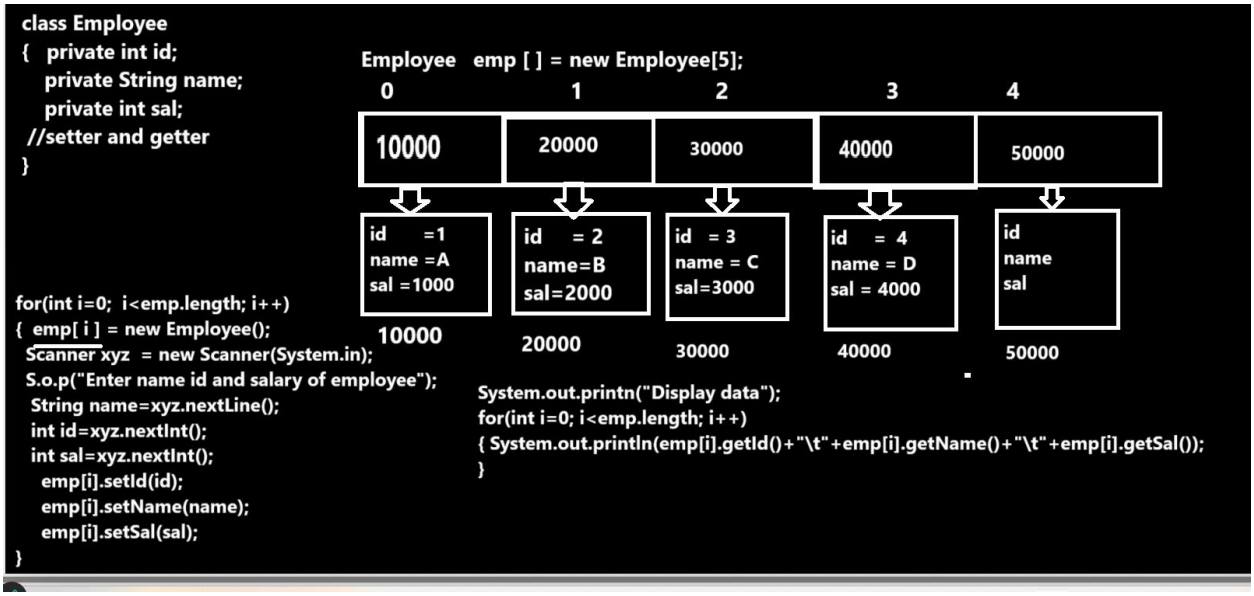
Array of objects is used for store more than one object using single name reference

Example: Suppose consider we have Employee class with field id name and salary and we want to create 5 objects of employee class and store its data and display so creating separate 5 objects is not good approach so better way we can create array of object of 5 employee class object

How to create array of object

Syntax: classname ref[]=new classname[size];//array of reference

```
for(int i=0;i<ref.length;i++){
  ref[i]=new classname(); //array of object
}
```



Example with source

```
import java.util.*;
class Employee
{ private int id;
  private String name;
  private int sal;

  public void setId(int i)
  { id=i;
  }
```

```
 public int getId(){
  return id;
 }
 public void setName(String n)
 { name=n;
 }
 public String getName()
 { return name;
 }
 public void setSal(int s)
 { sal=s;
 }
 int getSal()
 { return sal;
 }
}
public class ArrOfObjApp
{
  public   static void main(String x[])
  {
    Employee  e[]= new Employee[5];
          for(int i=0; i<e.length;i++)
          { Scanner xyz  = new Scanner(System.in);
            e[i]=new Employee();
                System.out.println("Enter name id and salary of employee");
                String name=xyz.nextLine();
                int id=xyz.nextInt();
                int sal=xyz.nextInt();
                e[i].setName(name);
                e[i].setId(id);
                e[i].setSal(sal);
          }
          System.out.println("Display employee data");
          for(int i=0;i<e.length;i++)
          {
           System.out.println(e[i].getName()+"\t"+e[i].getId()+"\t"+e[i].getSal());
          }
  }
```

}

Example: WAP to create class name as Product with field id, name and price and create array of object of 5 products and store data in it and display it as well as input the id from keyboard and search product present in array or not means search product by id if id found then show message product found otherwise show message product not found.

```
import java.util.*;
class Product
{  private int id;
   private String name;
   private int price;

   public void setId(int i)
   { id=i;
   }
   public int getId(){
     return id;
   }
   public void setName(String n)
   { name=n;
   }
   public String getName(){
     return name;
   }
   public void setPrice(int p)
   { price=p;
   }
   public int getPrice()
   { return price;
   }
}
public class ProdSearchApp
{
    public static void main(String x[])
          {Scanner xyz  = new  Scanner(System.in);

             Product p[]=new Product[5]; //array of reference
```

```
        for(int i=0;i<p.length;i++)
        {  p[i]=new Product(); //array of objects
                System.out.println("ente name id and price of product");
                String pname=xyz.nextLine();
                int pid=xyz.nextInt();
                int pprice=xyz.nextInt();
                p[i].setName(pname);
                p[i].setId(pid);
                p[i].setPrice(pprice);
                xyz.nextLine();
        }
        System.out.println("Display product details");
        for(int i=0; i<p.length;i++)
        {
          System.out.println(p[i].getId()+"\t"+p[i].getName()+"\t"+p[i].getPrice());
        }
        boolean flag=false;
        System.out.println("Enter product id for search");
        int pid=xyz.nextInt();
        for(int i=0; i<p.length;i++)
        {
                    if(p[i].getId()==pid)
                            { flag=true;
                      break;
                            }
        }
        if(flag)
        { System.out.println("Product found");
        }
        else
        { System.out.println("Product Not Found");
        }
      }
}
```

Example1: WAP to create class name as Student with field id, name and sub []
So you have create 5 objects of student class and store data in it and calculate the percentage of every student and display and student should give 6 subjects exam

Instance Variable/Static variable and local variable

---

a) If we declare any variable within class without static keyword called as instance variable and if we declare variable within class with static keyword called as static variable or class level variable and if we declare variable within class function called as local variable or any function called as local variable.

```
class ABC
{
    int x;  //instance variable
   static int y; // class level variable or static variable

  void show(int n ) //n is also local variable
  {
    int m=0; //local variable
  }
}
```

b) Instance variable can allocate memory after creating object of class and static variable can allocate memory before creating object of class and local variable can allocate after calling function.

Note: static variable allocate memory before creating object of class so we can access it without object i.e using class name and instance variable allocate memory after object means cannot access instance variable without creating object of class and function can allocate memory after calling so we cannot access it outside function block.

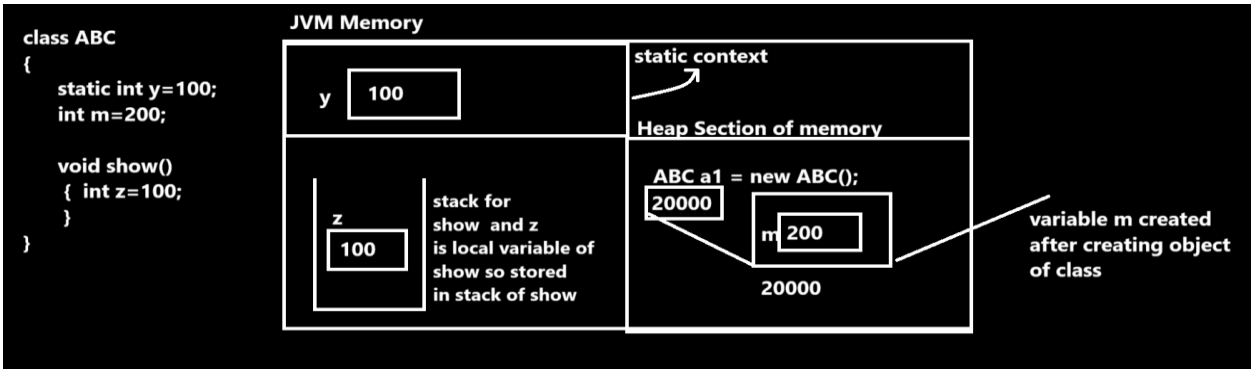
```
class ABC
{
   int x=200;
   static int y=100;

}
public class ABCAPP
{
    public static void main(String x[])
    {
       System.out.println("Y is  "+ABC.y);
       System.out.println("X is  "+ABC.x);

    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ABCAPP.java
ABCAPP.java:12: error: non-static variable x cannot be referenced from a static context
          System.out.println("X is  "+ABC.x);
                                         ^
1 error

C:\Program Files\Java\jdk1.8.0_291\bin>
```

error responsible code

Note: if we think about above code we get compile time error because we try to access the instance variable by using class name and it is not possible so we get compile time error

c) Instance variable can store in object and object stored in heap section of memory and static variable can store in permanent generation of memory or meta section of memory and local variable can store in stack of memory



Q. What is permanent generation section or Meta section after JDK 1.8?

PermGen stands for Permanent Generation. It is a special type of heap space. It is separate from the main memory (heap). JVM uses PermGen to keep track of loaded class metadata. All the static content is stored by the JVM to this memory section. Static content can be a static method, references to the static object and primitive variables. PermGen also contains information about bytecode, names, and JIT. Before releasing Java 7, String Pool is also available in this space and separate from it after releasing Java 7.

Q. What is heap memory section?

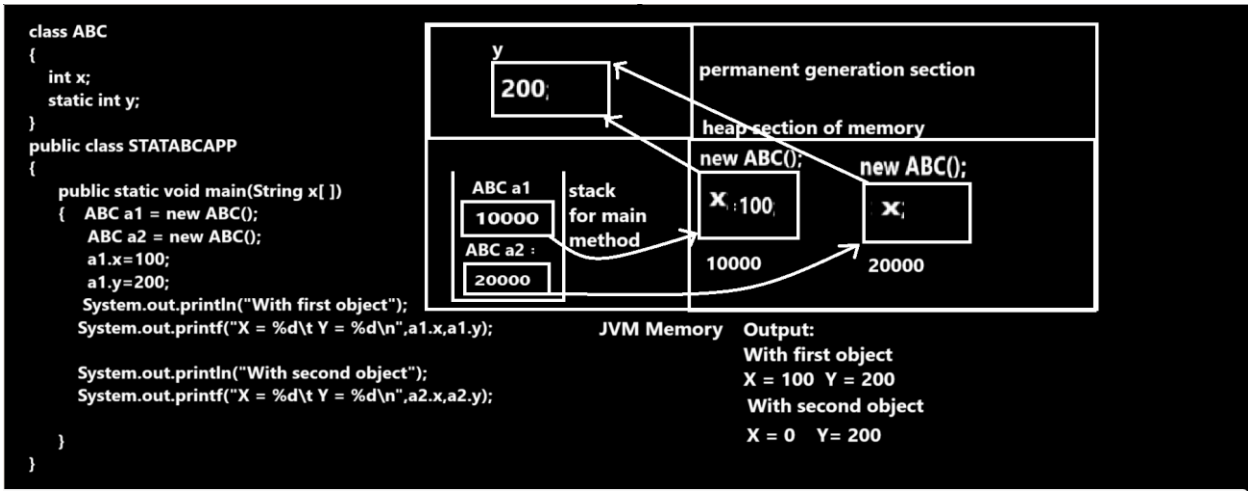
Heap memory is part of JVM memory and it is reserve by the JVM for allocate memory at run time for objects and array in java

Means when user use new keyword then JVM allocate block at program run time in heap section of memory and store data in it and if object or array not use by any reference means not use in future then JVM delete the memory of objects or array at run time called as garbage collection

Q. What is stack memory section?

Stack is temporary memory block created by JVM at program run time and it is create at the time of function calling and present in memory whenever function get execute when function definition get close the stack will delete from memory and all local variables of that function present in stack for temporary purpose and stack is also part of JVM memory

d) static variable can share its common copy between more than one object of same class and instance variable can share separate copy for every object and local variable can create on every function calling.



e) static variable present in memory till program in execution and instance variable present in memory when ever object present in memory

Note: before calculating life of instance variable we should know the life of object.

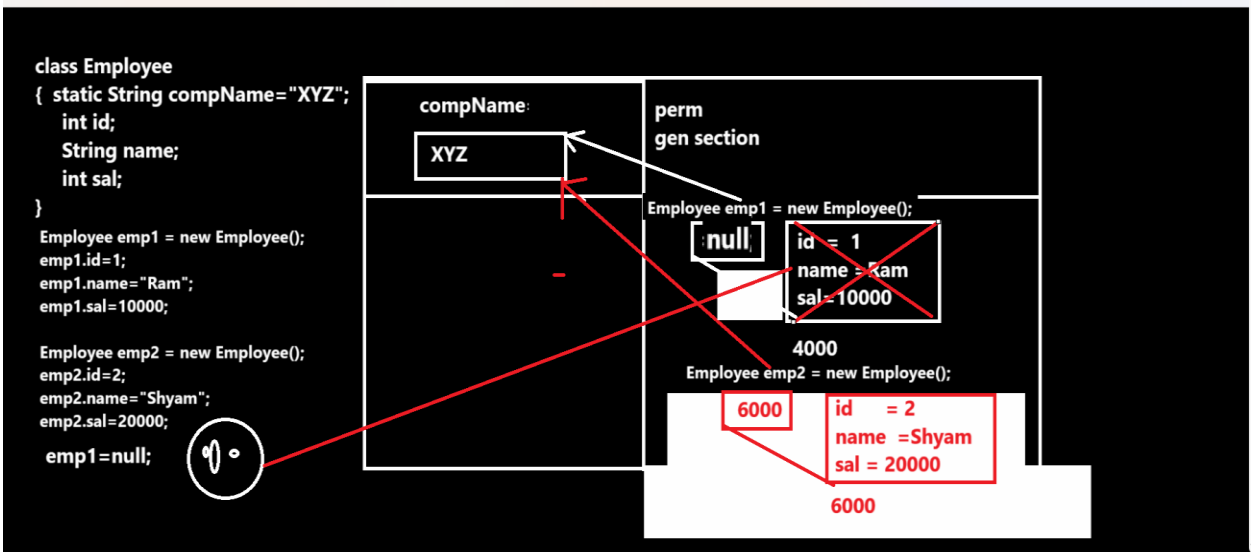
Q. Whenever object present in memory?

---

whenever object use by particular reference then object present in memory when object not use any reference then JVM automatically delete the memory of object called as garbage means when object delete by the JVM then instance variable also delete because instance variable is part of object.

static variable not delete from memory when object deleted by JVM because static variable is not part of object.

static variable share commonly in multiple object means if particular object deleted by JVM but for remaining object static must be present in memory



f) if we think about the static variable and instance variable has some default value provided by java to us according to his data type means when user not provide value them then JVM initialize default values to them

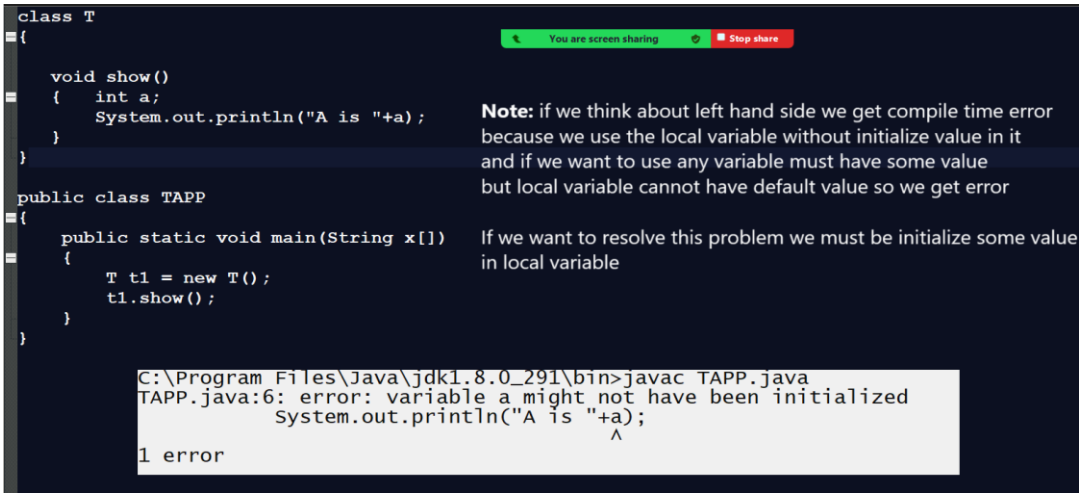
Data Type	Default value
int	0
float	0.0
Double	0.0
Long	0
Boolean	False
String	Null
Byte	0
Char	blank

Important point related with local variables

---

a. Local variable cannot access outside of function block and if we want to access values of local variable in outside of function we need to copy value of local variable in instance variable.

b. Local variable cannot have default value even garbage also means if we want to use local variable we must be initialize some value in it.



Example:

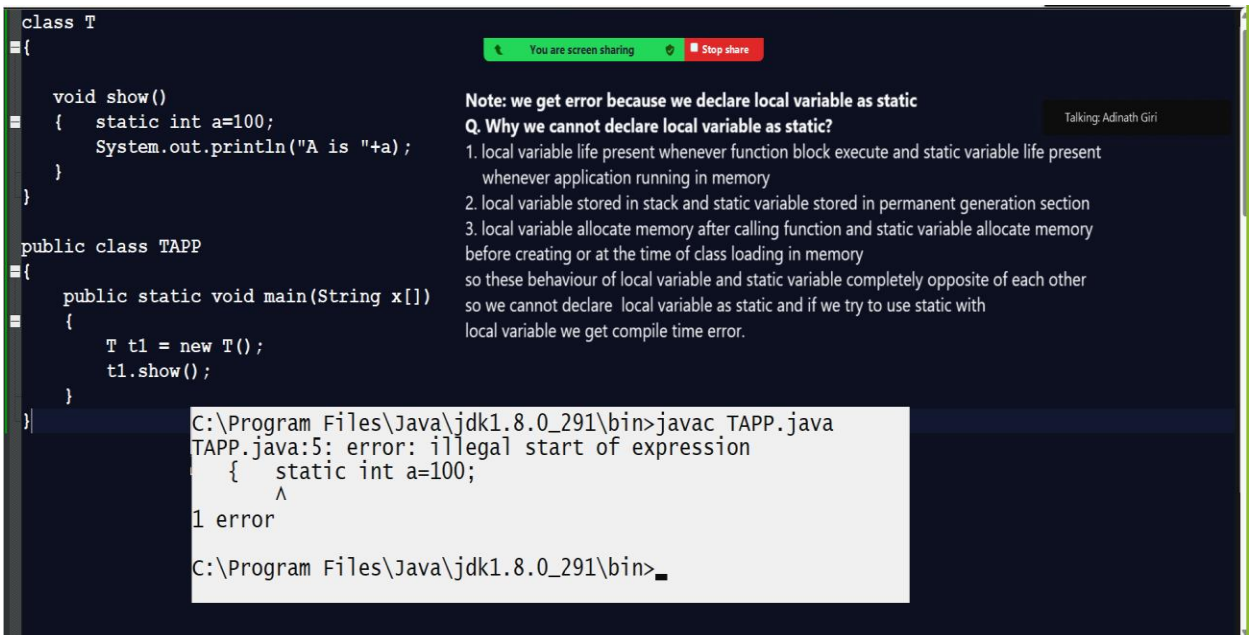
```
class T
{

  void show()
  {  int a=100;
```

```
        System.out.println("A is "+a);
  }
}

public class TAPP
{
   public static void main(String x[])
        {
              T t1 = new T();
              t1.show();
        }
}
```

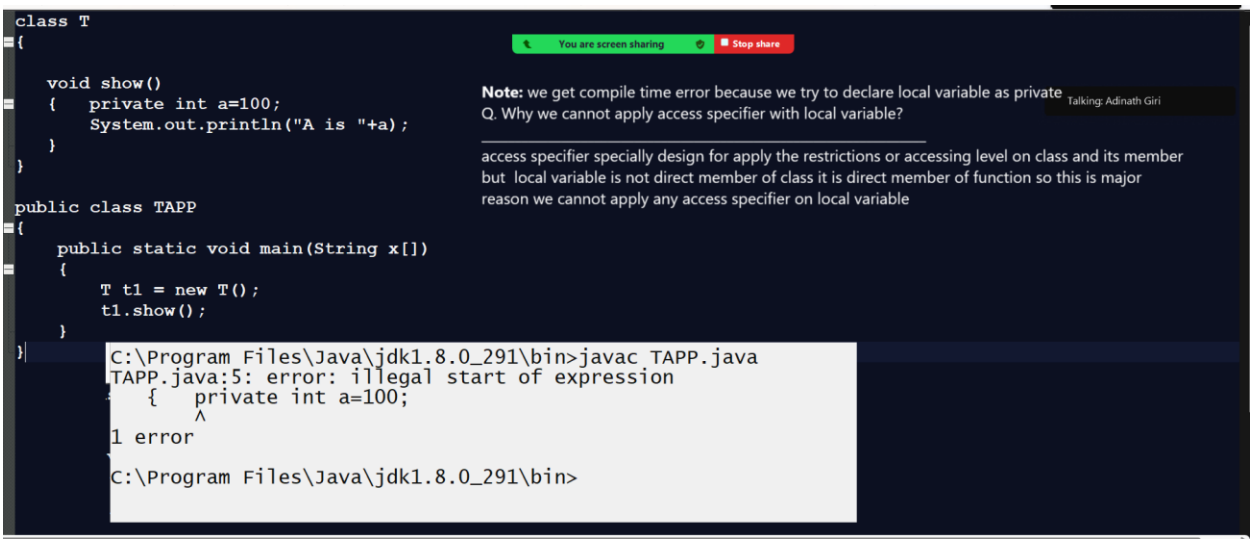
Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TAPP.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TAPP
A is 100

C:\Program Files\Java\jdk1.8.0_291\bin>
```

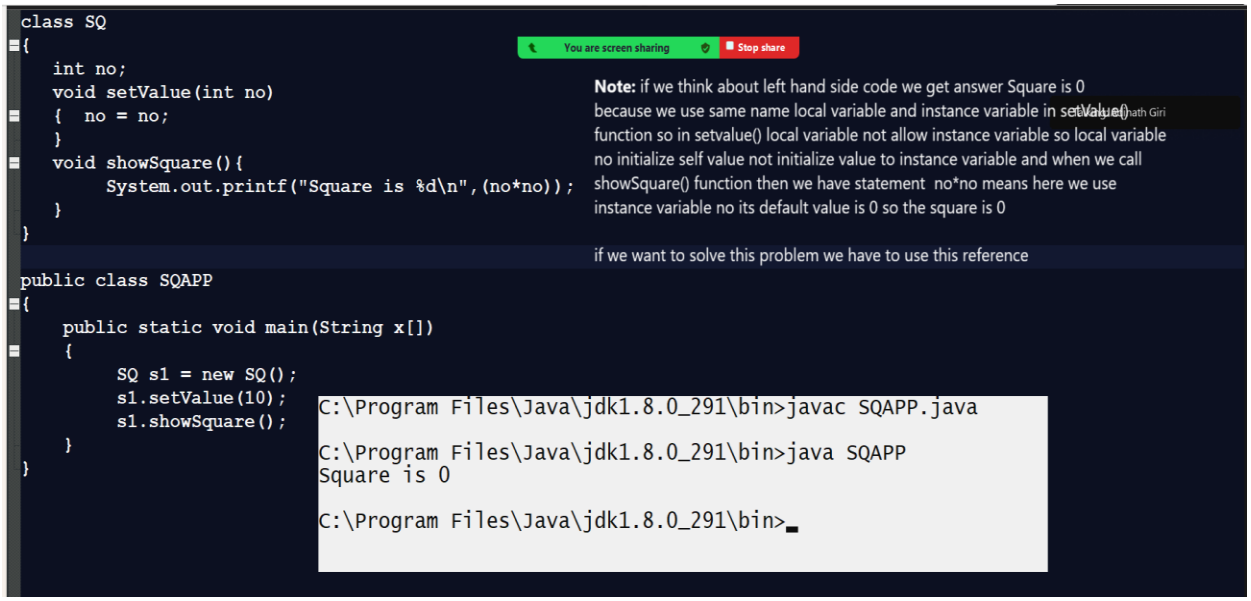
c) Local variable cannot declare as static and if we declare it as static then we get compile time error.



d) We cannot use any access specifier like as private, public or protected with local variable and if we try to use any access specifier with local variable we get compile time error.



e) If local variable name and instance variable name same then instance variable never access in local block shown in following code.

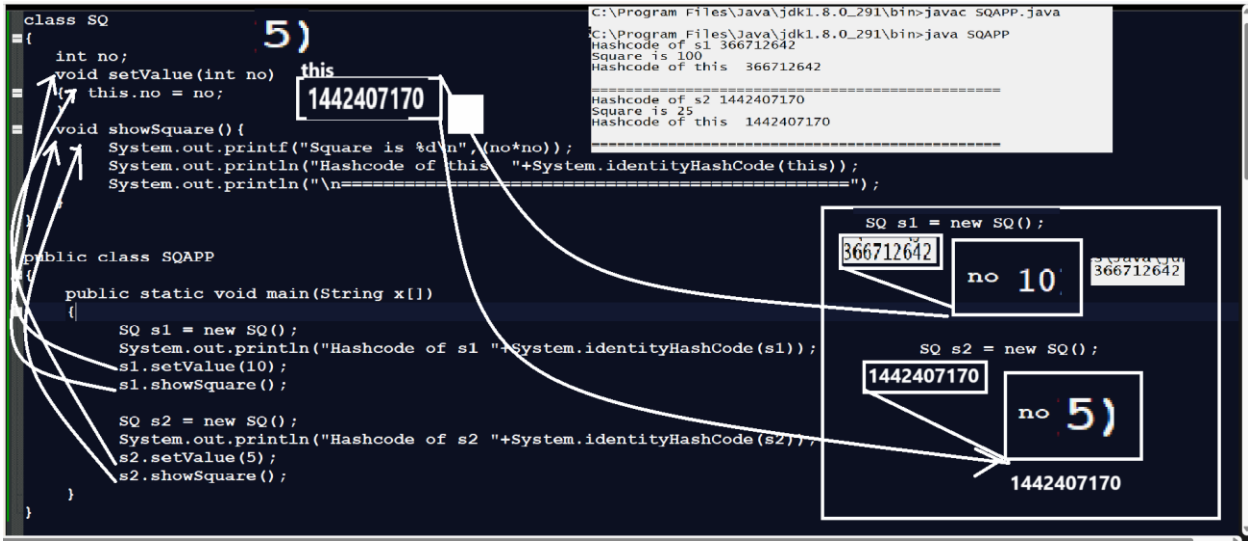


Q. What is this reference?

This is internal reference present in every class which is used for point to current running object in memory or working object in memory.

Q. How we can identify working object or current object?

Function calling object is working object or current running object in memory shown in following code and diagram.



Note: this reference recommended in two cases

a) When instance variable name and local variable name same

b) When we call same class member from its own definition

3. Provide Encapsulation

Q. What is encapsulation?

Encapsulation means hide the implementation detail from end user at implementation level or logical level called as encapsulation.

The goal of encapsulation is provide data security and we can achieve encapsulation by declaring class variable as private and access via public function.

So if we want to achieve encapsulation using java we need to declare class variable as private and access it via using public function

Means we can use POJO class for achieve encapsulation

Example of Encapsulation

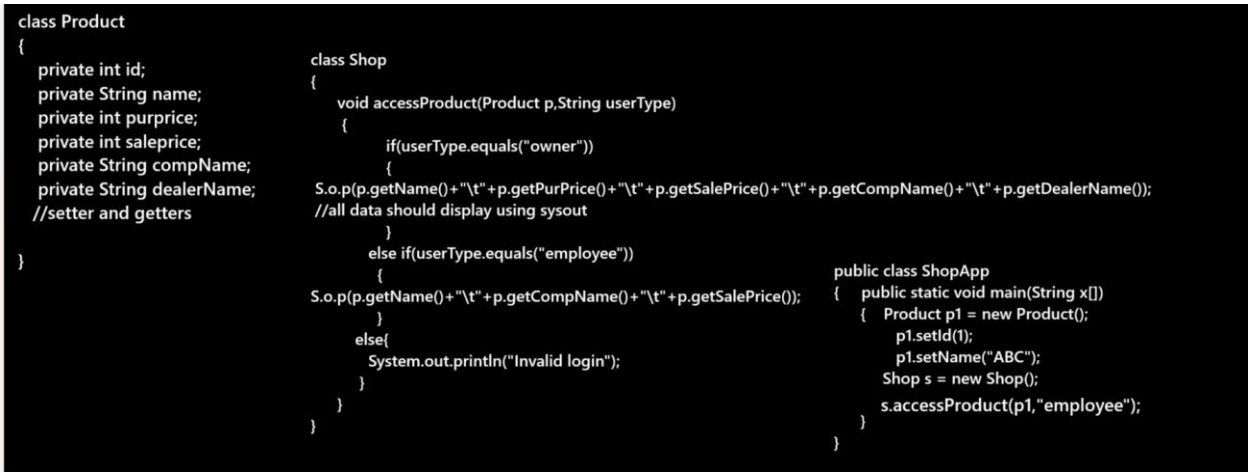
```
class Employee
{  private int id;
   private String name;

   public void setId(int id)
   { this.id=id;
   }
   public int getId(){
     return id;
   }
   public void setName(String name)
   { this.name=name;
   }
   public String getName(){
     return name;
   }
}
```

Q. Can we give real time scenario where I can use encapsulation?

Suppose consider we are working on Billing Application and we have some product database with field id, name, compname, mdate,edate, dealer,purpice,saleprice and we want to give some login to access the product details

Like as we want to give one login to shop keeper i.e owner and one login for employee

so we want to hide the some product details from employee like as purprice and dealername



```
class Product
{
   private int id;
   private String name;
   private int purprice;
   private int saleprice;
   private String compName;
   private String dealerName;
  //setter and getters

}
```

```
class Shop
{
     void accessProduct(Product p,String userType)
      {
            if(userType.equals("owner"))
            {
 S.o.p(p.getName()+"\t"+p.getPurPrice()+"\t"+p.getSalePrice()+"\t"+p.getCompName()+"\t"+p.getDealerName());
 //all data should display using sysout
            }
          else if(userType.equals("employee"))
           {
S.o.p(p.getName()+"\t"+p.getCompName()+"\t"+p.getSalePrice());
           }
         else{
           System.out.println("Invalid login");
          }
      }
}
```

```
public class ShopApp
{    public static void main(String x[])
     {   Product p1 = new Product();
          p1.setId(1);
          p1.setName("ABC");
        Shop s = new Shop();
        s.accessProduct(p1,"employee");
     }
}
```

Example with source code

```
class Product
{
  private int id;
  private String name;
  private String compName;
  private String dealerName;
  private int purPrice;
  private int salePrice;

  public void setId(int id)
  { this.id=id;
  }
  public int getId(){
    return id;
  }
  public void setName(String name)
  { this.name=name;
  }
```

```
  public String getName(){
    return name;
  }
  public void setCompName(String compName)
  { this.compName=compName;
  }
  public String getCompName(){
     return compName;
  }
  public void setDealerName(String dealerName)
  { this.dealerName=dealerName;
  }
  public String getDealerName(){
     return dealerName;
 }
  public void setPurPrice(int purPrice)
  { this.purPrice=purPrice;
  }
  public int getPurPrice(){
     return purPrice;
  }
  public void setSalePrice(int salePrice)
  { this.salePrice=salePrice;
  }
  public int getSalePrice(){
    return salePrice;
  }
}
class Shop
{
  void accessProduct(Product p1,String loginType)
  {
      if(loginType.equals("owner"))
            {

System.out.println(p1.getId()+"\t"+p1.getName()+"\t"+p1.getCompName()+"\t"+p1.getDealerN
ame()+"\t"+
            p1.getPurPrice()+"\t"+p1.getSalePrice());
```

```
            }
            else if(loginType.equals("employee"))
            {
System.out.println(p1.getId()+"\t"+p1.getName()+"\t"+p1.getCompName()+"\t"+p1.getSalePric
e());
            }
            else
            {
              System.out.println("Invalid login");
            }
  }
}
public class EncCapApp
{
  public static void main(String x[])
  {
                    Shop s1 =new Shop();

                    Product p1=new Product();
                    p1.setId(1);
                    p1.setName("ABC");
                    p1.setCompName("XYZ");
                    p1.setDealerName("STV");
                    p1.setPurPrice(10);
                    p1.setSalePrice(100);
                    s1.accessProduct(p1,"customer");
  }
}
```