ASSIGNMENT 8_1

1. **Why do we need static keyword in Java Programming?**

   ANS: A static keyword in java indicate that, the member is belongs to class.
   Static keyword in java mainly using for memory management.
   When we use static keyword in java then the memory that created for it it create only once.
   It can be use for making
   static variable, static method,static block
   for calling a static member its not mendotry to make object of that class, it can be acces by using its class name and also by using its object .

2. **What is class loading and how does the Java program actually executes?**

   Ans: class loading means that, loading a .class file for further process of execution of program its done by jvm.

   Execution of java program….
   1.source code compiled by comiler.
   2.once the code get compiled then compiler return or convert the source code into .class file that is also called bytecode
   3.after generating byte code the .class file get load by class loader sub system
   4.for loading class there having three step…

   1.loading:-in loading phase class loader locate the qualified .class file using its name, read class file and then convert it into class object. Class object contain the metadata of class file such as constructer,method.

   2.linking: In linking phase jvm perform many operation like allocating memory to the different variable of class, verifying format of code etc.

   3.initilizing:in initialization phase jvm initialize the default value to the variable and after that run the static block if any.

3. **Can we mark a local variable as static?**

   Ans: No

4. **Why is a static method also called a class method?**

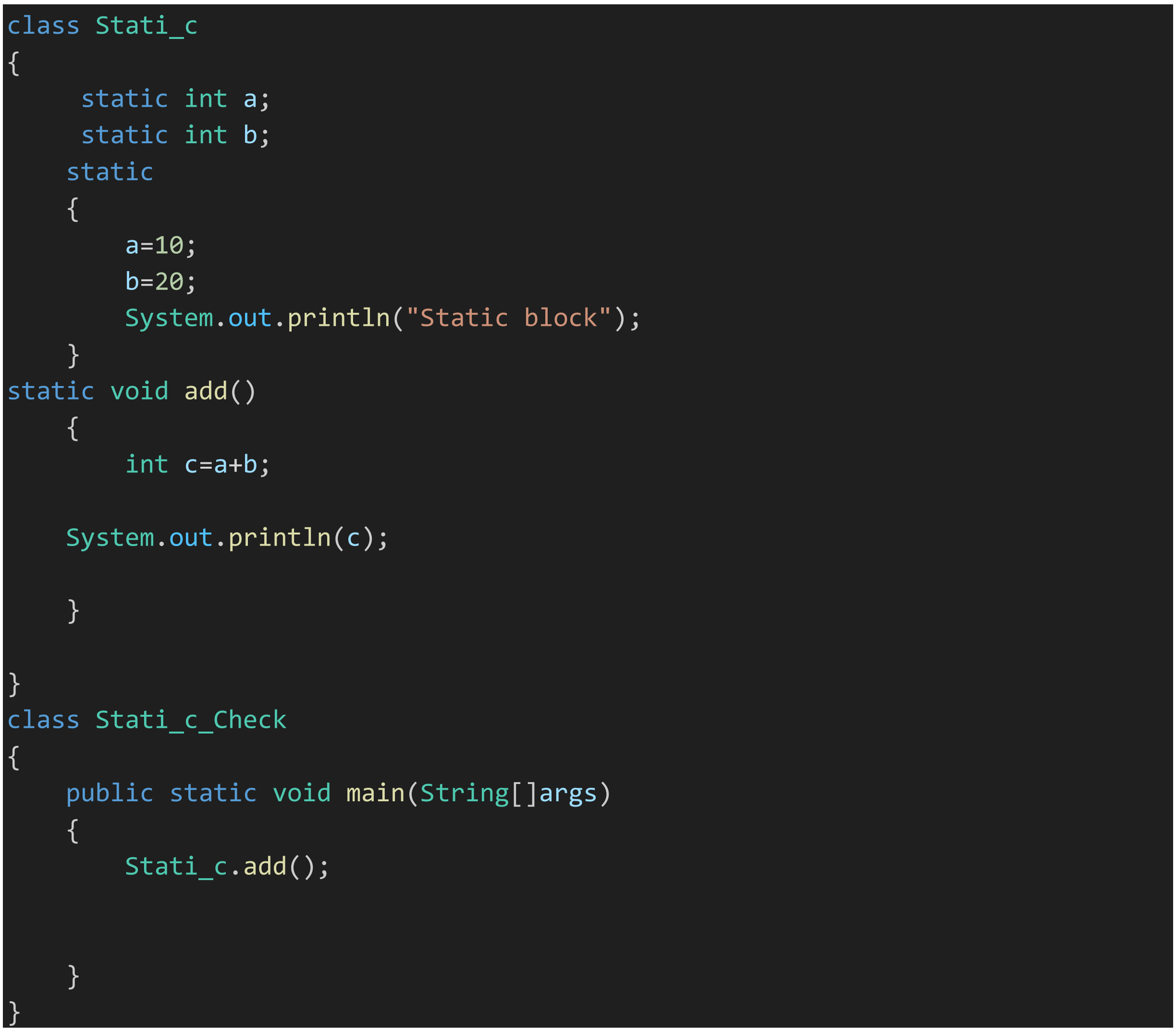
   Ans: because a static method can be also called by its class name rather than making by making object of that class.

5. **Why is the static block executed before the main method in java?**

   Ans: because in java when the .class file get loaded into the memory then jvm first execute the static block, after that main metod get executed.

6. **What is the use of static keyword Explain with an example.**

   Ans: by using static keyword we can save memory .
   Static keyword indicate that the member is belongs to the class. For calling them its not mendotry to make the object of class.



```
class Stati_c
{
     static int a;
     static int b;
    static
    {
        a=10;
        b=20;
        System.out.println("Static block");
    }
static void add()
    {
        int c=a+b;

    System.out.println(c);

    }

}
class Stati_c_Check
{
    public static void main(String[]args)
    {
        Stati_c.add();


    }
}
```

7. **The difference between a static variable and instance variable is as follows:**

<u>Static variable:</u>

1. The memory allocation of static variable get places after loading of class.
2. When we use static variable then its memory allocate only once.
3. Static variable are created by using static keyword.
4. Its also called class variable,
5. Its get stored in method area.
6. If the value doesn’t change from object to object then we should use static variable.
7. Static variable can be use inside the instance method, static method,static block, inner static class.

<u>Instance carriable:</u>

1. The memory allocation of instance variable get places after making object of class.
2. Its also called instance variable,

3. If the value change from object to object then we should use instance variable.

8. Difference between static and non static members of a class

8. **Ans:** The memory allocation of static variable get places after loading of class.
9. When we use static variable then its memory allocate only once.
10. Static variable are created by using static keyword.
11. Its also called class variable,
12. Its get stored in method area.
13. If the value doesn’t change from object to object then we should use static variable.

<u>Instance carriable:</u>

4. The memory allocation of instance variable get places after making object of class.
5. Its also called instance variable,

6. If the value change from object to object then we should use instance variable.

Pratical question asmt 8.2:

1.Create a class that keeps track of the number of instances created. Implement a static variable and method to accomplish this.

Ans:

2.Write a program and create a constructor with parameters and initialise the variable using a constructor.

Ans: Ans:

```
class  Asmt_8_2
{
    int num1;
    int num2;
    int num3;
    Asmt_8_2(int a,int b, int c)
    {
        num1=a;
        num2=b;
        num3=c;
    }
    int add()
    {
        return num1+num2+num3;
       // System.out.println("Addition of hree number is :"+res);

    }

}
class LaunchAsmt8_2
{
    public static void main(String[] args)
    {
        Asmt_8_2 obj1 = new Asmt_8_2(10,20,30);

       int res;
       res= obj1.add();
       System.out.println("the result of additin is"+res);

    }
}
```

3.Use a private keyword for a variable and use setter and getter methods to initialise and print the values.

Ans:

```
class Sett_erGetter

{
     private int age;
```

```
    private String name;

    // using setter for initilze variables.....

    public void setData(int age,String name)
    {
        this.age=age;
        this.name=name;
    }
    // using getter for accesing data....

    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
    }
}
class Sett_erGetterAsmt8_3
{
    public static void main(String[] args)
     {
        Sett_erGetter obj1 = new Sett_erGetter();
        obj1.setData(18,"vipin.v");
        String a;
        a=obj1.getName();
        int b;
        b=obj1.getAge();
        System.out.println("Name is:"+a);
        System.out.println("age is:"+b);


    }
}
```

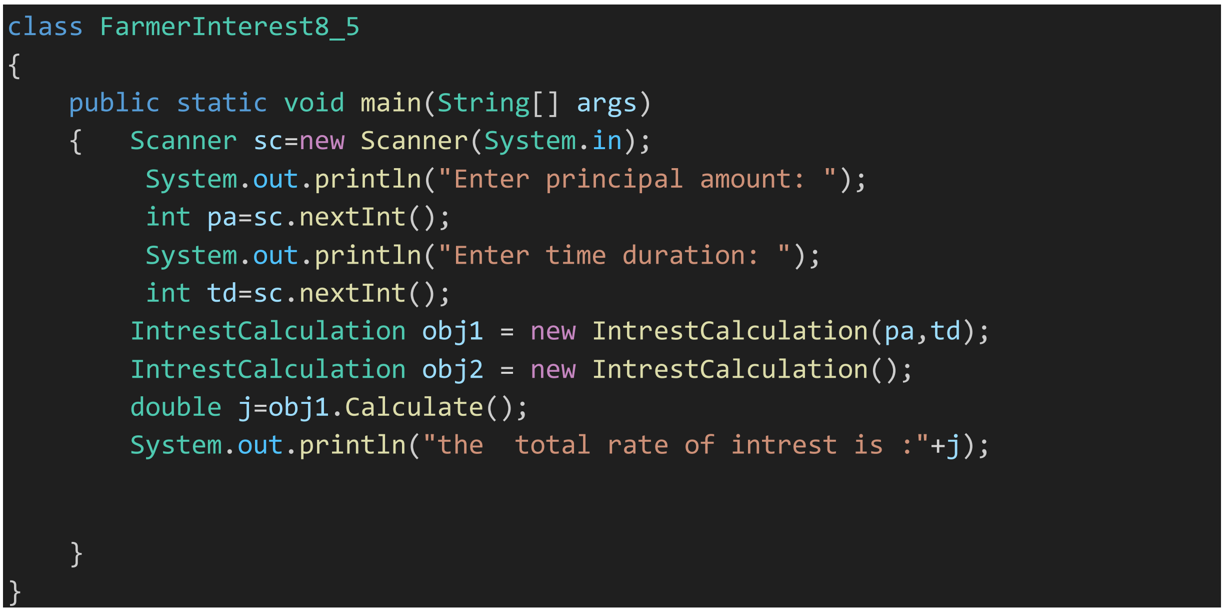
4.Write a program to call an method without creating an object of a class

Ans:
```
class StaticNonStatic

{
    static int i;
    static int j;
    static
    {
        System.out.println("this is static block.");
        i=1;
        j=2;

    }
    static void add()
    {
        System.out.println("this is static function.");
        System.out.println("static block"+(i+j));

    }

}
class AbcStaticExecution
{
 public static void main(String[] args) {
    System.out.println("this is main method");
    StaticNonStatic.add();

}
}
```

5.Write a program which has static block and constructor overloading,initialise variable using constructors and print it.

Ans:

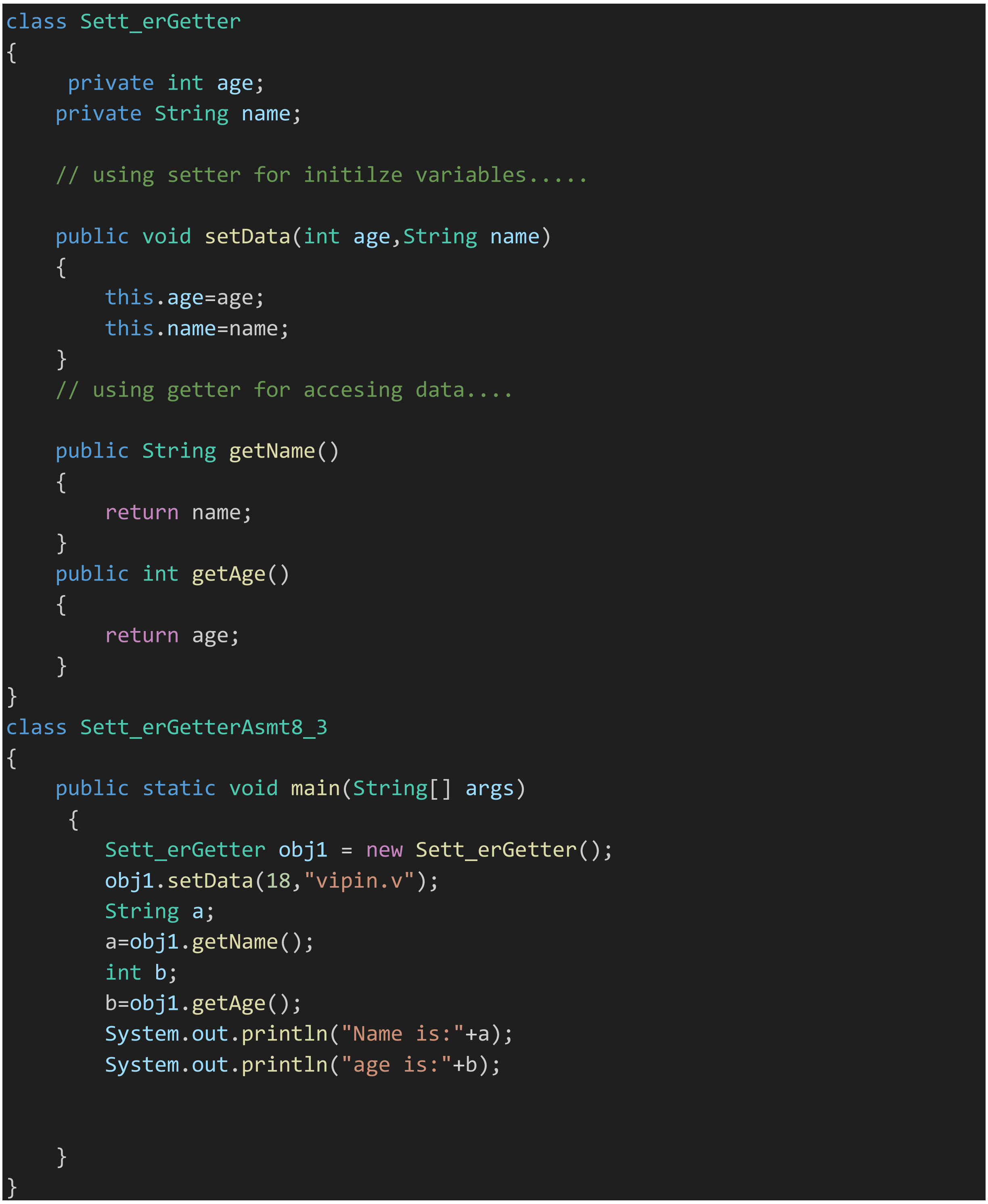
```
import java.util.Scanner;

class IntrestCalculation
{
    static int percent;
      double r;
    int p;
    int t;
    double i;
    static
    {
        //assigning valu to static variable..
        percent=100 ;

    }
    IntrestCalculation(int a,int b) // parameterized constructer
    {
        p=a;
        t=b;
        r=2.5/100*percent;
    }
    IntrestCalculation() // default constructer
    {
         Calculate();
    }
    public double Calculate()//method of calculation.
    {
        return p*r*t;
    }


}
```



```
class FarmerInterest8_5
{
    public static void main(String[] args)
    {   Scanner sc=new Scanner(System.in);
         System.out.println("Enter principal amount: ");
         int pa=sc.nextInt();
         System.out.println("Enter time duration: ");
         int td=sc.nextInt();
        IntrestCalculation obj1 = new IntrestCalculation(pa,td);
        IntrestCalculation obj2 = new IntrestCalculation();
        double j=obj1.Calculate();
        System.out.println("the  total rate of intrest is :"+j);


    }
}
```

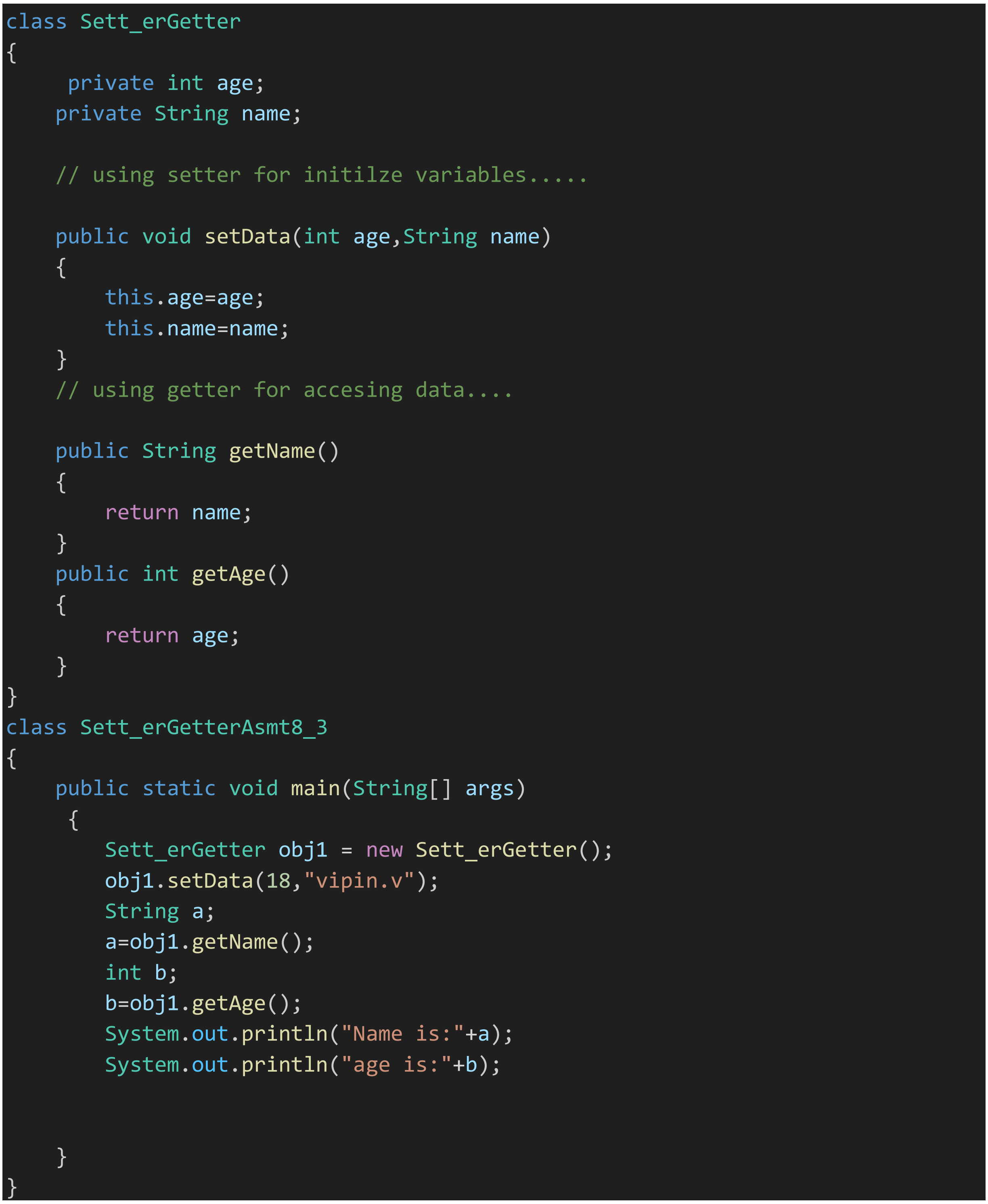
ASSIGNMENT 8-3

1. **What is Encapsulation in Java? Why is it called Data hiding?**
   ANS: in java encapsulation is the process of binding data and its corresponding method into the single unit .
   Encapsulation are called data hiding because once when we use this technique for hiding data (method , variable) then we cant access them from other class.
2. **What are the important features of Encapsulation?**
   Ans:
3. **What are getter and setter methods in Java Explain with an example.**
   Ans : getter and setter are two method in java that are using for setting and accessing data .



```java
class Sett_erGetter
{
     private int age;
    private String name;

    // using setter for initilze variables.....

    public void setData(int age,String name)
    {
        this.age=age;
        this.name=name;
    }
    // using getter for accesing data....

    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
    }
}
class Sett_erGetterAsmt8_3
{
    public static void main(String[] args)
     {
        Sett_erGetter obj1 = new Sett_erGetter();
        obj1.setData(18,"vipin.v");
        String a;
        a=obj1.getName();
        int b;
        b=obj1.getAge();
        System.out.println("Name is:"+a);
        System.out.println("age is:"+b);


    }
}
```

4. **What is the use of this keyword explain with an example**
   Ans: it is using to differenciate between instance variable and local variable or parameter that have the same name.
   Here are an example to illustrate it in detailed……



```
class Sett_erGetter
{
     private int age;
    private String name;

    // using setter for initilze variables.....

    public void setData(int age,String name)
    {
        this.age=age;
        this.name=name;
    }
    // using getter for accesing data....

    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
    }
}
class Sett_erGetterAsmt8_3
{
    public static void main(String[] args)
     {
        Sett_erGetter obj1 = new Sett_erGetter();
        obj1.setData(18,"vipin.v");
        String a;
        a=obj1.getName();
        int b;
        b=obj1.getAge();
        System.out.println("Name is:"+a);
        System.out.println("age is:"+b);


    }
}
```

5.What is the advantage of Encapsulation?

Ans: 1. Encapsulation allow us to do modification easily in program.
2.it provide data binding in single unit that is class.
3.data get protected when we use this technique means we can not access the peivate data from another class
4. it provide maintainability of program**.**

6. How to achieve encapsulation in Java? Give an example.

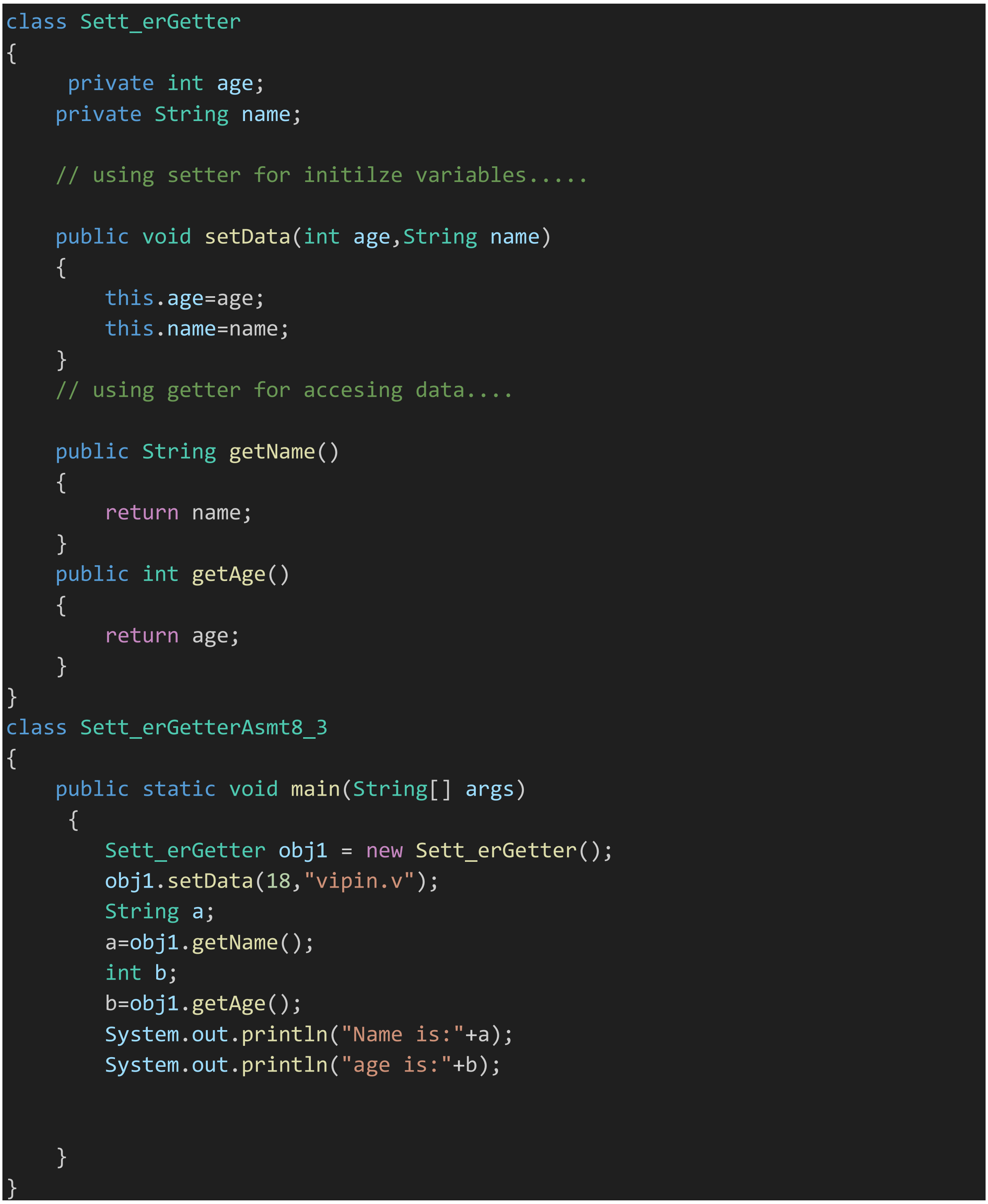
Ans: In java we can achieve encapsulation by declaring private to the instance variable of class.

And to modify these variable setter and getter method is used .

This provide a control to access of these private variable

Using object of that class rather than accessing variable directly from outside the class

Here are the example ….



```
class Sett_erGetter
{
     private int age;
    private String name;

    // using setter for initilze variables.....

    public void setData(int age,String name)
    {
        this.age=age;
        this.name=name;
    }
    // using getter for accesing data....

    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
    }
}
class Sett_erGetterAsmt8_3
{
    public static void main(String[] args)
     {
        Sett_erGetter obj1 = new Sett_erGetter();
        obj1.setData(18,"vipin.v");
        String a;
        a=obj1.getName();
        int b;
        b=obj1.getAge();
        System.out.println("Name is:"+a);
        System.out.println("age is:"+b);


    }
}
```

In the above programme I try to demonstrate the use of encapsulation in java

For that In program having two method thas name start with is setter and getter these variable are working here to assign the value and accesing them respectly.

Assignment 8_4

1. **What is a Constructor?**
   Ans: Constructer is special member function that name are same like class name.
   a constructer is like method that are using for initializing the object and like method it can have a collection of statement.
   A constructer can be parameterized and non parameterized a constructer doesn't hold a any return type not even void. It return instance value.

2. **What is Constructor Chaining?**
   Ans: A constructer chaining is a technique in which we can jump one constructer to another constructer by using this().

3. **Can we call a subclass constructor from a superclass constructor?**
   Ans: No. There is no way in java to call a subclass constructor from a superclass constructor**.**
4. **What happens if you keep a return type for a constructor?**
   Ans: ideally a constructer should not have a return type.
   If we give return type to it then it will treated as method**.**
5. **What is No-arg constructor?**
   Ans: no argument constructer means a constructer that not take argument means that not have parameter.
   This type of constructer mainly created by compiler which called default constructer or it can be also user defined.
   Example

   ```
   Class Demo
   {
   Const_ructer()
   {
   //no argument constructer.
   }
   }
   ```
6. **How is a No-argument constructor different from the default Constructor?**

   Ans: **No-argument constructor**

   When a program have any parameterized *constructer* then it is also mendotry to define no argument constructer because once the programmer define any constructer in program then compiler not going to make any constructer by it self so in that case no argument constructer if need then it also need to define by programmer.

default Constructor?

Default constructer define by compiler at runtime.

It doesn’t hold any parameter.

7. **When do we need Constructor Overloading?**
   Ans: some times there are need more than one constructer for initialize the object. And when we use more than one constructer in program then that is called constructer overloading

```
class Xyz
{
    int a;
    int b;
    int c=0;
    Xyz(int d,int e)
    {
        a=d;
        b=e;
    }
    Xyz(int e,int f, int g)
    {
        a=e;
        b=f;
        c=g;
    }
    public int add()
    {
        return a+b+c;
    }
}
class Constructer_Overloading
{
public static void main(String[] args) {
    Xyz obj1 = new Xyz(1,2);

   int g1=obj1.add();
   System.out.println(g1);
   int g2;
   Xyz obj2 = new Xyz(1,2,3);

 g2=obj2.add();
     System.out.println(g2);



}

}
```

8. **What is Default Constructor?**

Ans: in java constructer are responsible for initialize the obect .
So in java when programmer not make any constructer then the java compiler make it by itself .
Default constructer doesn't hold any parameter.