Day-2 6/5/25

class Demo { static{ System.out.println("Static Block"); } Demo(){ //Method property also class name so niether class nor method that is Constructor System.out.println("Constructor"); } } public class q2{ public static void main(String[] args) { Demo d1 = new Demo(); Demo d2 = new Demo();

}

}

Output Static Block ERROR! error: can't find main(String[]) method in class: Demo

CONSTRUCTORS :

Constructors are used to initialize the optics, for very class there will be a default constructor

whenever we create an object, constructor will be called then object will be created.

- constructor having class name and methods properties, so neither class nor method.

example:

in this we created two objects so two times constructor will be called

```
public class q2{
   public static void main(String[] args){
      Demo d1 = new Demo()
      Demo d2 = new Demo()
   }
}
```

STATIC BLOCk :

static block will be created with the static keyword, it will called automatically no need to call it with either object or class.

STATIC METHOD:

For static method no need to call with object you can directly call with class name and here static will act as instance or object.

- Instance/ local

if we have age then it will be local variable but the different ages is called instance variable.

```
public class Person {
    // Instance variable
    private int age;

    // Setter method
    public void setAge(int age) {
        this.age = age;
    }

    // Getter method
    public int getAge() {
        return age;
    }


    public static void main(String[] args) {
        Person p = new Person();
        p.setAge(25);
        System.out.println("Age is: " + p.getAge());
    }
}
```

with main class

```
class Main{
    public static void main(String args[]){
        Person obj = new Person();
        obj.setAge(24);
        int res= obj.getAge();
        System.out.println(res);
    }
}
class Person{
    int age;
    void setAge(int age1){
    this.age = age1;
}
int getAge(){
    return age;
```

```
}
}
```

for String name

```
string name;
void setName(String name) {
        this.name = name;
}
String getName() {
        return name;
    }
```

with main class

```
class Main{
    public static void main(String args[]){
        Person obj = new Person();
        obj.setName("Sam");
        String res= obj.getName();
        System.out.println(res);
    }
}
class Person{
    String name;
    void setName(String name){
        this.name = name;
    }
String getName(){
    return name;

}
}
```

for bool passedout

```
Boolean passedout;
setPassedOut(boolean passedOut) {
        this.passedOut = passedOut;
    }
boolean isPassedOut() {
```

```
        return passedOut;
    }
```

with main class

for float marks

```
float marks;
void setMarks(float){
    this.marks = marks;
}
float getMarks() {
        return marks;
    }
```

for long ERP

```
long erp;
void setErp(){
    this.erp = erp;
}
long getErp(){
    return erp;
}
```

RECURSION:

Fabonacci Series :

For best cases we have mention in if loop as [**if(n==0) return 0] [else f(n-1)+ f(n-2)]**

```
class Main{
    public static void main(String[] args){
        Fibonacci obj = new Fibonacci();
        int res = obj.fib(7);
        System.out.println(res);
    }
}
class Fibonacci{
    int fib(int n){
```

```
        if(n>2){
            return n;
        }
        else{
            return fib(n-1) + fib(n-2);
        }
    }
}
```

Factors :

```
class Main {
    public static void main(String[] args) {
        Factor obj = new Factor();
        obj.printFactors(28);
    }
}

class Factor {
    void printFactors(int n) {
        System.out.println("Factors of " + n + " are:");
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                System.out.println(i);
            }
        }
    }
}
```

LCM & HCF:

```
class Main {
    public static void main(String[] args) {
        LcmHcf obj = new LcmHcf();
        int a = 24;
        int b = 36;

        int hcf = obj.findHCF(a, b);
        int lcm = obj.findLCM(a, b);

        System.out.println("HCF of " + a + " and " + b + " is: " + hcf);
        System.out.println("LCM of " + a + " and " + b + " is: " + lcm);
```

```
    }
}

class LcmHcf {
    int findHCF(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }

    int findLCM(int a, int b) {
        return (a * b) / findHCF(a, b);
    }
}
```

CONDITIONAL STATEMENTS:

if, else, and else if ladder and switch case

conditional statements are used to check the condition and to print the relavant block in contant time.

example sudo code:

if :

```
int number = 10;
if(number>10){
   System.out.println("positive");
}
```

else :

```
int n=10;
if(n>0){
    System.out.println("positive");
}
else{
    System.out.println("negative");
}
```

else if ladder :

```
int n=10;
if(n>0){
    int n=10;
if(n>0){
    System.out.println("positive");
}
else if{
    System.out.println("negative");
}
else{
    System.out.println("zero");

}
}
```

switch case :

```
int day = 3;
String dayName;

switch (day) {
    case 1:
        dayName = "Monday";
        break;
    case 2:
        dayName = "Tuesday";
        break;
    case 3:
        dayName = "Wednesday";
        break;
    default:
        dayName = "Invalid day";
}

 System.out.println(dayName);
```

CONTROL STATEMENTS:

Enter control loops: for , while

Exit control loops:

1. Entry Control loop

For:

```
class Person{
   int[] arr = {1,2,3,4,5};
   for(int i = 0;i<n;i++){
      return arr[i];
   }
   for(start;end;diff){     //syntax

   }
}
```

example of for loop (table of 5)

```
class Person {
   public static void main(String[] args) {
      int n = 5; // The number for the table
      for (int i = 1; i <= 10; i++) {
         System.out.println(n + " * " + i + " = " + (n * i));
      }
   }
}
```

While:

```
class Person{
   int[] arr = {1,2,3,4,5};
   for(int i = 0;i<n;i++){
      return arr[i];
   }
   start;
   while(end){  //syntax
       diff;
   }
}
```

example of while loop (table of 5)

```
class Person {
```

```
    public static void main(String[] args) {
        int number = 5; // The number for the table
        int i = 1; // Initialize the counter

        while (i <= 10) {
            System.out.println(number + " * " + i + " = " + (number * i));
            i++; // Increment the counter
        }
    }
}
```

when we know the range we prefered use for loop and when we don’t know the range prefered to use while loop.

Reverse a number using divide and modulus (/,%):

we can use this for array burn if enterd the value which is not there we use (%) as n%arr so we don’t get array.

```
while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number = number / 10;
        }

        System.out.println("Reversed number: " + reversed);
    }
}
```