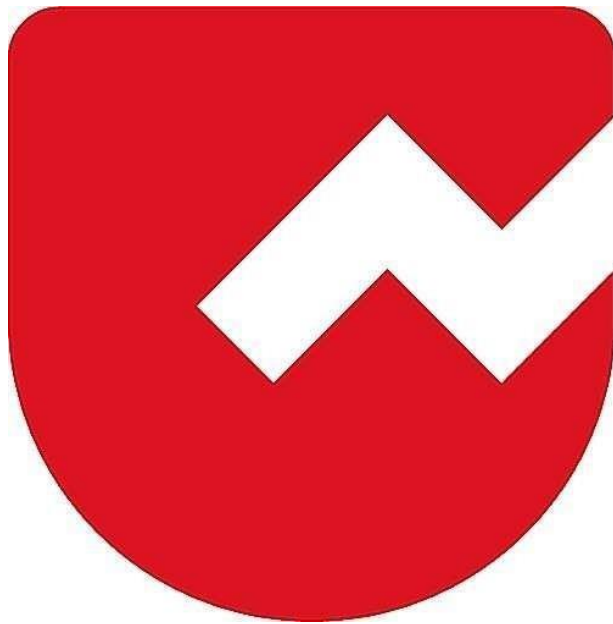




Zoho Schools for Graduate Studies



Notes

SCOPE AND LIFE TIME OF VARIABLE

1. What is the output of the following code?

```
class First{  
    int a;  
    public static void main(String[] args){  
        System.out.println(a);  
    }  
}
```

Ans: Compiler Error

Reason: Non-Static Instance Variable cannot be accessed directly inside the static context.

2. What is the output of the following code?

```
class First{  
    int a;  
    public static void main(String[] args){  
        First f;  
        System.out.println(f.a);  
    }  
}
```

Ans: Compiler error

Reason: Object is not even created just Object name is given.

3. What is the output of the following code?

```
class First{  
    int a;  
    public static void main(String[] args){  
        First f = new First();  
        System.out.println(f.a);  
    }  
}
```

Ans: 0

Reason: Instance Variable if uninitialized gets default initialization.

4. What is the output of the following code?

```
class First{  
    int b;  
    public static void main(String[] args){  
        int a;  
        First f = new First();  
        System.out.println(a);  
    }  
}
```

Ans: Compiler Error

Reason: Local Variable (a) is not initialized

5. What is the output of the following code?

```
class First{  
    int a;  
    public static void main(String[] args){  
        int a = 2;  
        System.out.println(a);  
    }  
}
```

Ans: 2

Reason: Local Variable can have same name as that of Instance Variable but bad programming practice. In Main method we initialized local variable so it gets printed.

6. What is the output of the following code?

```
class First{  
    int a;  
    public static void main(String[] args){  
        int a = 2;  
        First f = new First();  
        System.out.println(f.a);  
    }  
}
```

Ans: 0

Reason: Here the print statement uses instance variable. Instance variable if uninitialized will have the default value.

7. What is the output of the following code?

```
class First{  
    public static void main(String[] args){  
        int a;  
        First f = new First();  
        a = 3;  
        f.go();  
        System.out.println(a);  
    }  
    void go(){  
        int x = a;  
        System.out.println(x);  
    }  
}
```

Ans: Compiler Error

Reason: In go() method x is initialized with (a) but a is not initialized in that scope.

SCOPE: The region in a program where upto which the variable is accessible or visible.

LIFE TIME:

- How long the variable exists before it is destroyed.
- Destroying refers to deallocating the memory that was allotted to the variables when declaring it.

Variable Type	Scope	Lifetime
Instance variable	Throughout the class except in static methods	Until the object is available in the memory
Class variable	Throughout the class	Until the end of the program
Local variable	Within the block in which it is declared	Until the control leaves the block in which it is declared

Local Variable:

- Accessible within the block.
- Destroyed outside the block.
- Allocated in Stack memory.
- Access modifiers cannot be used for local variables.
- Final is the only non-access modifier used.

Instance Variable:

- Accessible within the class(Methods,blocks and constructors).
- Visibility of instance variable can be controlled using access modifiers.
 - Public – Can access outside the class also
 - Protected – Subclasses of same and different packages

- Private – only in that class.
- Generally it is recommended to use private
- Allocated in Heap memory
- Out of 8 Non-access modifiers only static, final and transient are applicable to instance variable.
- Life time: Until the object is available in the memory.

Static Variable:

- Accessible throughout the class
- Only one copy of static variable per class exist
- Shared by all objects created from the class
- Used to refer common property for all objects
- Allocated in class area or static memory area
- Get memory at the time of class loading
- Destroyed when the program terminates
- Scope is similar to instance variable
- Generally it is recommended to declare as public

8. What is the output of the following code?

```
class First{
    int a;
    public static void main(String[] args){
        int a;
        First f = new First();
        a = 3;
        f.go();
        System.out.print(a); }
}
```

```
void go(){
    int x = a;
    System.out.print(x);
}
}
```

Ans: 03

Reason: In the method go() instance variable is used whereas in the main method local variable is used.

9. What is the output of the following code?

```
class First{
    int a;
    public static void main(String[] args){
        int a;
        First f = new First();
        a = 3;
        f.go();
        System.out.print(a+f.a);
    }
    void go(){
        int x = a;
        a = 3;
        System.out.print(x);
    }
}
```


Ans: 06

Reason: In the method go() instance variable is assigned a value and in the main method both variables are used.

10. What is the output of the following code?

```
do{  
  
    int x = 0;  
  
    System.out.println(x++);  
  
}while(x<=10);
```

Ans: Compiler Error

Reason: In the condition we used x but x is local variable. So in dowhile loop declare and initialize the variable out of the dowhile loop.

11. What is the output of the following code?

```
int x;  
  
do{  
  
    x = 0;  
  
    System.out.println(x++);  
  
}while(x<=10);
```

Ans: 0 Prints infinitely.

Reason: x is initialized everytime while the loop runs.

12. What is the output of the following code?

```
int x=0;  
  
do{  
  
    System.out.print(x++);  
  
}while(x<=10);
```

Ans: 012345678910

Reason: Initialization done outside the do while and just the incrementation takes place inside the loop.