Variable scope
Best practices

If you follow the 2 rules below, you will not get the bugs discussed in the following pages.

1. Declare all your variable at the top of the program
   ⇒ No undeclared variables

2. Good to also initialize the variables when you declare them
   ⇒ No uninitialized variables
Bug - Variable Scope –

variable declared only in one branch of if

We need to make sure that we have a single variable, called **result**, that is alive:
- at the if part.
- at the else part.
- at the **printf** statement at the end.

```java
import java.util.Scanner; // incorrect code
public class Example
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.printf("Please enter an integer: ");
        int number = in.nextInt();

        if (number >= 0)
        {
            // this result is not the same as the one in else
            // this one is only 'alive' in these { }
            int result = number * number;
        }
        else
        {
            // result is undeclared
            result = number * 10; // result is undeclared
        }
        // result is undeclared
        System.out.printf("result = %d\n", result);
    }
}
```
Bugs - Variable Scope – variable declared only in branch(es) of if

Samples of correct code

```java
int result = -1;
/* This result is visible in all the code from now on in this method (or block).
   It was initialized so even though there is no else branch, for that case it will still have a value (-1).

if (number >= 0)
{
    result = number * number;
}
System.out.printf("result = %d\n",result);
```

```java
int result;
/* This result is visible in all the code from now on in this method (or block).
   It was NOT initialized but both possible paths (on the if-branch or the else-branch) initialize it.

if (number >= 0)
{
    result = number * number;
}
else
{
    result = number * 10;
}
System.out.printf("result = %d\n",result);
```

```java
int result;
/* This result is visible in all the code from now on in this method (or block).

if (number >= 0) {
    int result = number * number;
}
else {
    int result = number * 10; // undeclared
}
System.out.printf("result = %d\n",result); //undeclared
```

Samples of incorrect code

```java
if (number >= 0) {
    result = number * number; // undeclared
}
else {
    result = number * 10; // undeclared
}
System.out.printf("result = %d\n",result); //undeclared
```

```java
if (number >= 0) {
    int result = number * number;
}
else {
    int result = number * 10; // undeclared
}
System.out.printf("result = %d\n",result); //undeclared
```

```java
if (number >= 0) {
    int result = number * number; // undeclared
}
else {
    int result = number * 10; // undeclared
}
System.out.printf("result = %d\n",result); //undeclared
```

```java
if (number >= 0) {
    result = number * number; // undeclared
}
else {
    result = number * 10; // undeclared
}
int result = -1;
System.out.printf("result = %d\n",result);
```

We need to make sure that we have a single variable, called result, that is alive:

• at the if part.
• at the else part.
• at the printf statement at the end.