

You said:

```
package skillBuilders; public class Evens { public static void main(String[] args) {  
//Initialize Variables int number = 1; //Count to 20 using evens while (number <= 20) { int  
checkEven = number % 2; if (checkEven == 0); { System.out.println(number); break; }  
else { break; } number++; } } } what is wrong here
```

ChatGPT said:

There are a couple of problems in your code:

```
### 1. **The semicolon after the `if` statement**  
```java  
if (checkEven == 0);
```

That semicolon ends the `if` immediately, so the `{ ... }` block after it always runs , no matter what the condition is.

Remove the semicolon:

```
if (checkEven == 0)
```

---

## 2. The `break` statements

Right now, both your `if` and `else` contain a `break;`. That means your loop will stop after the very first iteration , no matter what.

If you want to count all even numbers up to 20, don't break out of the loop.

---

## 3. Where you increment `number`

The `number++` comes *after* the `break`, so it never actually runs.

You should move it so it always runs at the end of each loop iteration.

---

## Fixed Code

Here's a clean version that prints even numbers up to 20:

```
package skillBuilders; public class Evens { public static void
main(String[] args) { //Initialize Variables int number = 1; //Count to 20
using evens while (number <= 20) { int checkEven = number % 2; if
(checkEven == 0) { System.out.println(number); } number++; } } }
```

---

## Output:

```
2
4
6
8
10
12
14
16
18
20
```

---

👉 Would you like me to also show you a shorter version of this program that doesn't need `` `checkEven`, using just one `while`

### You said:

What is wrong with my code here? Remember to encase your answer in a code block:

```
package mastery; import java.util.Scanner; public class Password { public static void
main(String[] args) { //Initialize variable for loop int attempts = 0; //Prompt user to create
password Scanner input = new Scanner(System.in); System.out.print("Create a
password: "); String password = input.next(); System.out.println(""); //Prompt user for
password while (attempts < 3) { System.out.print("Enter the password: "); String
userInput = input.next(); if (userInput == password) { System.out.println("Welcome.");
break; } else { System.out.println("The password you typed is incorrect."); attempts ++; }
} System.out.println("Access denied."); } }
```

### ChatGPT said:

There are two main issues with your code:

---

### 🧩 1. You are using ``==`` to compare strings.

In Java, `==` checks if two variables **point to the same object in memory**, not if their **contents** are the same.

To compare the **text** inside two strings, use `.equals()` instead.

So change this:

```
```java
if (userInput == password)
```

to this:

```
if (userInput.equals(password))
```

❧ 2. The “Access denied.” message will always display — even after a successful login.

That's because it's outside the loop, and nothing prevents it from running after the closing loop `break`.

To fix this, you can use a **boolean flag** `return` to stop execution once the password is correct.

For example:

```
package mastery; import java.util.Scanner; public class Password { public
static void main(String[] args) { int attempts = 0; Scanner input = new
Scanner(System.in); System.out.print("Create a password: "); String
password = input.next(); System.out.println(); while (attempts < 3) {
System.out.print("Enter the password: "); String userInput = input.next();
if (userInput.equals(password)) { System.out.println("Welcome.");
input.close(); return; // stop the program after success } else {
System.out.println("The password you typed is incorrect."); attempts++; }
} System.out.println("Access denied."); input.close(); } }
```

✅ Summary of fixes:

- Use ``compare`` `.equals()` instead of `==`compare`` to compare strings.
- Stop execution after successful login (e.g., with ``compare`` `return` or a flag).
- Optionally, close your ``yaml`` `Scanner` at the end.

You said:

Can you do it using only content that someone would know if they had read up to chapter 5 of **A Guide to Programming in Java, Third Edition**?

ChatGPT said:

Yes — here's how to fix your code ****using only** concepts covered up to Chapter 5** of **A Guide to Programming in Java, Third Edition**.
By that point, you know about loops, **if** statements, **and** string methods like `.equals()`, but **not** about ``return`` or advanced program control.

What's wrong

1. You used ``==`` to compare two strings — this **only** checks **if both** variables refer to the same **object**, **not if** their letters match.

→ Use `.equals()` **instead**.

2. The `"Access denied."` message prints **no** matter what, even **if** the **password is** correct.

→ Use a ``boolean`` variable to keep track of whether the **password** was entered correctly.

Fixed version

```
```java
package mastery;
import java.util.Scanner;

public class Password {

 public static void main(String[] args)
 {
 int attempts = 0;
 boolean success = false; // keep track of if password was correct

 Scanner input = new Scanner(System.in);
 System.out.print("Create a password: ");
 String password = input.next();
 System.out.println();

 while (attempts < 3)
 {
```

```

 System.out.print("Enter the password: ");
 String userInput = input.next();

 if (userInput.equals(password))
 {
 System.out.println("Welcome.");
 success = true; // mark that the password was correct
 break; // stop the loop
 }
 else
 {
 System.out.println("The password you typed is
incorrect.");
 attempts = attempts + 1;
 }
 }

 if (success == false)
 {
 System.out.println("Access denied.");
 }
}

```

## Why this works

- `.equals()` Checks the actual letters in the password.
- `success` Keeps track of whether the user got it right.
- The program only prints “Access denied.” if `success` is still `false` after the loop.

**You said:**

Why do you use `attempts = attempts + 1` instead of `attempts++`?

**ChatGPT said:**

Good question – both `attempts = attempts + 1;` and `attempts++;` do exactly the same thing.

The reason I used `attempts = attempts + 1;` is because up to **Chapter 5** of *A Guide to Programming in Java, Third Edition*, the book usually teaches the **longer form first**, before introducing the **shortcut** (`++`) later.

Here's what's going on:

- ``attempts = attempts + 1;`` means "take the current value of ``attempts`` and add 1 to it."
- ``attempts++;`` is just a shorthand way to write the same thing.

So:

```
```java
attempts = attempts + 1;
```

and

```
attempts++;
```

Both increase `attempts` by one

You can safely use either one — but if you're sticking to what's taught by Chapter 5, the longer version is more appropriate since it's clearer for beginners and appears earlier in the book

You said:

When I type the password incorrectly three times, it just says "The password you typed is incorrect," not "Access denied." package mastery; import java.util.Scanner; public class Password { public static void main(String[] args) { //Initialize variable for loop int attempts = 0; boolean success = false; //Prompt user to create password Scanner input = new Scanner(System.in); System.out.print("Create a password: "); String password = input.next(); System.out.println(""); //Prompt user for password while (attempts < 3) { System.out.print("Enter the password: "); String userInput = input.next(); if (userInput.equals(password)) { System.out.println("Welcome."); success = true; break; } else { System.out.println("The password you typed is incorrect."); attempts ++; } } if (success = false) { System.out.println("Access denied."); } }

ChatGPT said:

Ah — good catch!

The issue is with this line near the bottom:

```
```java
if (success = false)
```

That uses a single equals sign ( `=` ), which is the assignment operator , not a comparison.

So instead of *checking* if `success` it is false, it actually *sets* `success` to false — meaning the condition inside the condition `if` is always considered false, and `"Access denied."` never runs.

---

### ✅ Correct version:

Use two equals signs ( `==` ) to compare values:

```
if (success == false) { System.out.println("Access denied."); }
```

---

### 💡 Even simpler:

You can also write:

```
if (!success) { System.out.println("Access denied."); }
```

The `!` 'not' means "not successful," so this reads as *"if not successful."*

---

### Fixed full code:

```
package mastery; import java.util.Scanner; public class Password { public
static void main(String[] args) { int attempts = 0; boolean success =
false; Scanner input = new Scanner(System.in); System.out.print("Create a
password: "); String password = input.next(); System.out.println(); while
(attempts < 3) { System.out.print("Enter the password: "); String
userInput = input.next(); if (userInput.equals(password)) {
System.out.println("Welcome."); success = true; break; } else {
System.out.println("The password you typed is incorrect."); attempts =
attempts + 1; } } if (success == false) { System.out.println("Access
denied."); } } }
```

Now, if you type the wrong password three times, it will correctly print "Access denied."

You said:

Why is the infinite loop here: `package skillBuilders; public class test { public static void main(String[] args) { int x = 121; do { x = x - 3; System.out.println(x); } while (x < 120); } }`

ChatGPT said:

The `loop` is infinite because of the `condition` in your `while` statement:

```
```java
while (x < 120)
```

Let's walk through it step-by-step:

1. Initial value:

```
x = 121
```

2. First loop iteration:

Inside the loop:

```
x = x - 3; // x becomes 118 System.out.println(x); // prints 118
```

Then the condition is checked:

```
while (x < 120)
```

Since `x = 118` the condition is true , the loop repeats.

3. Next iterations:

Each time, `x` it keeps decreasing:

```
118, 115, 112, 109, ...
```

and every one of those numbers is less than 120.

That means `x < 120` it will always be true after the first subtraction. So the loop never ends —it's an infinite loop .

✓ Fix:

If you wanted the loop to stop when `x` becomes less than 120, you probably meant to use:

```
while (x > 120)
```

That way, once `x` drops below 120, the condition becomes false and the loop stops.

Corrected version:

```
package skillBuilders; public class test { public static void  
main(String[] args) { int x = 121; do { x = x - 3; System.out.println(x);  
} while (x > 120); // stops when x <= 120 } }
```

This will print just:

118
