# Code Assessment

1. **Answer:**

To improve the clarity, maintainability, and robustness of the code, consider the following best practices:

1. **Use Enum for Card Types:** Define an enumeration for card types instead of using string literals. This makes the code more readable and avoids potential typos.
2. **Use Switch Statement:** Replace the series of if-else statements with a switch statement. This can improve readability when checking multiple cases.
3. **Encapsulate Logic in Methods:** If the logic for accepting or rejecting a transaction is more complex, consider encapsulating it in separate methods. This promotes modularity and easier testing.
4. **Handle Unknown Card Types:** Always consider handling unexpected or unknown card types gracefully. Logging an error or throwing an exception might be appropriate.

By applying these best practices, the code becomes more readable, maintainable, and extensible. Additionally, it helps avoid potential bugs and provides better support for future changes or additions to card types.

Best Practice code will be:

public enum CardType {

PREMIUM,

MASTER,

YOUTH

}

public class ATM {

public boolean processTransaction(Card card) {

switch (card.getType()) {

case PREMIUM:

case YOUTH:

return false;

case MASTER:

return acceptTransaction();

default:

log.error("Unknown card type: " + card.getType());

return false;

}

}

private boolean acceptTransaction() {

// Accept transaction logic

return true;

}

}

1. **Answer**

public class Main {

public static void main(String[] args) {

int n = 0;

while (n < 10) {

int a = n + 2;

int b = a + n;

System.out.println("a = " + a);

System.out.println("b = " + b);

n++; // Increment n to eventually exit the loop

}

System.out.println("n = " + n);

}

}

**Changes made:**

1. Corrected the capitalization of **While** to **while**.
2. Moved the print statements for 'a' and 'b' outside the loop to be inside. Note that variables declared inside the loop have limited scope and are not accessible outside the loop.
3. replaced **n = 5;** with **n++;** inside the loop. This will increment the value of **n** in each iteration, allowing the loop to eventually exit when **n** becomes greater than or equal to **10**.