

Below are the implementations and discussions based on your requirements:

---

### 1. For Loop to Print Even Numbers from 1 to 20

```
for (int i = 1; i <= 20; i++) {  
    if (i % 2 == 0) {  
        System.out.println(i);  
    }  
}
```

---

### 2. While Loop to Prompt User for Flight Choice Until a Valid Number is Entered

```
import java.util.Scanner;  
  
public class FlightChoice {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int choice = -1;  
  
        while (choice < 1) {  
            System.out.println("Enter your flight choice (a positive number): ");  
            if (scanner.hasNextInt()) {  
                choice = scanner.nextInt();  
                if (choice < 1) {  
                    System.out.println("Invalid choice. Try again.");  
                }  
            } else {  
                System.out.println("Invalid input. Please enter a number.");  
                scanner.next(); // Clear invalid input  
            }  
        }  
  
        System.out.println("You selected flight: " + choice);  
    }  
}
```

```
}  
}
```

---

### 3. Pros and Cons of Different Loops

#### For Loop

- **Pros:** Simple when iterating over a range or array with a known size.
- **Cons:** Less flexible for conditional or dynamic iteration.

#### While Loop

- **Pros:** Ideal for indefinite iterations until a condition is met.
- **Cons:** May lead to infinite loops if the condition is not properly managed.

#### Do-While Loop

- **Pros:** Guarantees at least one execution of the loop body.
  - **Cons:** Not always intuitive; the condition check occurs after execution.
- 

### 4. Fibonacci Sequence using a For Loop

```
public class Fibonacci {  
    public static void main(String[] args) {  
        int n1 = 0, n2 = 1, n3;  
        System.out.print(n1 + " " + n2);  
  
        for (int i = 2; i < 10; i++) {  
            n3 = n1 + n2;  
            System.out.print(" " + n3);  
            n1 = n2;  
            n2 = n3;  
        }  
    }  
}
```

---

### 5. Sum of Integers from 1 to 100 Using a While Loop

```
public class SumWhileLoop {
```

```
public static void main(String[] args) {  
    int sum = 0, i = 1;  
    while (i <= 100) {  
        sum += i;  
        i++;  
    }  
    System.out.println("Sum: " + sum);  
}  
}
```

---

## 6. Do-While Loop Prompting Until Negative Number is Entered

```
import java.util.Scanner;
```

```
public class DoWhileExample {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int number;  
  
        do {  
            System.out.println("Enter a number (negative to exit): ");  
            number = scanner.nextInt();  
        } while (number >= 0);  
  
        System.out.println("Exited loop.");  
    }  
}
```

---

## 7. Use of Continue Statement in a Loop

```
public class ContinueExample {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {
```

```
        if (i == 5) {  
            continue;  
        }  
        System.out.println(i);  
    }  
}  
}
```

---

## 8. Initialize and Print a 2D Array

```
public class TwoDArray {  
    public static void main(String[] args) {  
        int[][] array = {  
            {1, 2, 3},  
            {4, 5, 6},  
            {7, 8, 9}  
        };  
  
        for (int i = 0; i < array.length; i++) {  
            for (int j = 0; j < array[i].length; j++) {  
                System.out.print(array[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

---

## 9. For vs. While Loops

- Use **for loops** when the number of iterations is known.
  - Use **while loops** for conditional or indefinite iterations.
- 

## 10. Break Statement in Nested Loops

**Impact:**

- Exits the closest enclosing loop.

**Considerations:**

- Can make code harder to follow.
  - Use with clear documentation to avoid confusion.
- 

**11. Arrays in Java****Example:**

```
int[] numbers = {1, 2, 3, 4, 5};  
for (int num : numbers) {  
    System.out.println(num);  
}
```

---

**12. Differences in Memory Allocation**

- Single-dimensional: Continuous memory blocks.
  - Multidimensional: Array of references.
- 

**13. Uninitialized Arrays****Pitfalls:**

- NullPointerException.
  - Must be explicitly initialized.
- 

**14. Find Maximum Value in Array**

```
public static int findMax(int[] array) {  
    int max = array[0];  
    for (int num : array) {  
        if (num > max) {  
            max = num;  
        }  
    }  
    return max;  
}
```

```
}
```

---

### 15. Average of Numbers in an Array

```
public static double findAverage(int[] array) {  
    int sum = 0;  
    for (int num : array) {  
        sum += num;  
    }  
    return (double) sum / array.length;  
}
```

---

### 16. Sum of 2D Array

```
public static int sum2DArray(int[][] array) {  
    int sum = 0;  
    for (int[] row : array) {  
        for (int num : row) {  
            sum += num;  
        }  
    }  
    return sum;  
}
```

---

### 17. Find Minimum and Maximum in an Array

```
public static int[] findMinMax(int[] array) {  
    int min = array[0], max = array[0];  
    for (int num : array) {  
        if (num < min) min = num;  
        if (num > max) max = num;  
    }  
    return new int[]{min, max};  
}
```

---

## 18. Static vs. Dynamic Arrays

### Static Arrays

- Fixed size.

### Dynamic Arrays

- Flexible size.

### Drawbacks:

- Performance overhead for resizing.
- 

## 19. Merge Two Sorted Arrays

```
public static int[] mergeSortedArrays(int[] arr1, int[] arr2) {  
    int[] result = new int[arr1.length + arr2.length];  
    int i = 0, j = 0, k = 0;  
  
    while (i < arr1.length && j < arr2.length) {  
        if (arr1[i] < arr2[j]) {  
            result[k++] = arr1[i++];  
        } else {  
            result[k++] = arr2[j++];  
        }  
    }  
  
    while (i < arr1.length) result[k++] = arr1[i++];  
    while (j < arr2.length) result[k++] = arr2[j++];  
  
    return result;  
}
```

---

## 20. Reverse an Array

```
public static void reverseArray(int[] array) {  
    int left = 0, right = array.length - 1;
```

```
while (left < right) {  
    int temp = array[left];  
    array[left] = array[right];  
    array[right] = temp;  
    left++;  
    right--;  
}  
}
```

---

## 21. Find Second Largest Element

```
public static int findSecondLargest(int[] array) {  
    int largest = Integer.MIN_VALUE, secondLargest = Integer.MIN_VALUE;  
    for (int num : array) {  
        if (num > largest) {  
            secondLargest = largest;  
            largest = num;  
        } else if (num > secondLargest && num != largest) {  
            secondLargest = num;  
        }  
    }  
    return secondLargest;  
}
```

---

## 22. First Even Number in List

```
for (int num : list) {  
    if (num % 2 == 0) {  
        System.out.println(num);  
        break;  
    }  
}
```

---



### 23. Print Odd Numbers Using Continue

```
for (int i = 1; i <= 20; i++) {  
    if (i % 2 == 0) {  
        continue;  
    }  
    System.out.println(i);  
}
```

---

### 24. Prompt Until Negative Number

```
Scanner scanner = new Scanner(System.in);  
  
int number;  
  
do {  
    System.out.println("Enter a number: ");  
    number = scanner.nextInt();  
} while (number >= 0);
```

### 25. Multiplication Table Skipping Multiplication by 5

```
public class SkipMultiplesOfFive {  
    public static void main(String[] args) {  
        int number = 7; // Example: Multiplication table for 7  
        for (int i = 1; i <= 10; i++) {  
            if (i == 5) {  
                continue;  
            }  
            System.out.println(number + " x " + i + " = " + (number * i));  
        }  
    }  
}
```

---

### 26. Count from 1 to 10 but Break at 6

```
public class BreakAtSix {
```

```
public static void main(String[] args) {  
    for (int i = 1; i <= 10; i++) {  
        if (i == 6) {  
            break;  
        }  
        System.out.println(i);  
    }  
}  
}
```

---

### **27. Skip Number 5 While Counting from 1 to 10**

```
public class SkipFive {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {  
            if (i == 5) {  
                continue;  
            }  
            System.out.println(i);  
        }  
    }  
}
```

---

### **28. Check if a Number is Prime**

```
import java.util.Scanner;
```

```
public class PrimeNumberCheck {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Enter a number: ");  
        int number = scanner.nextInt();  
        boolean isPrime = true;
```

```

if (number <= 1) {
    isPrime = false;
} else {
    for (int i = 2; i <= Math.sqrt(number); i++) {
        if (number % i == 0) {
            isPrime = false;
            break;
        }
    }
}

if (isPrime) {
    System.out.println(number + " is a prime number.");
} else {
    System.out.println(number + " is not a prime number.");
}
}

```

---

## 29. Reverse the Digits of a Number

```

import java.util.Scanner;

public class ReverseDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number: ");
        int number = scanner.nextInt();
        int reversed = 0;

        while (number != 0) {

```

```
        int digit = number % 10;

        reversed = reversed * 10 + digit;

        number /= 10;
    }

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

---

### 30. Multiplication Table for a Given Number

```
import java.util.Scanner;

public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a number: ");
        int number = scanner.nextInt();

        System.out.println("Enter the range: ");
        int range = scanner.nextInt();

        for (int i = 1; i <= range; i++) {
            System.out.println(number + " x " + i + " = " + (number * i));
        }
    }
}
```

---

### 31. Count Vowels and Consonants

```
import java.util.Scanner;

public class VowelConsonantCount {
    public static void main(String[] args) {
```

```

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string: ");

String input = scanner.nextLine().toLowerCase();

int vowels = 0, consonants = 0;

for (char c : input.toCharArray()) {
    if ("aeiou".indexOf(c) != -1) {
        vowels++;
    } else if (Character.isLetter(c)) {
        consonants++;
    }
}

System.out.println("Vowels: " + vowels);
System.out.println("Consonants: " + consonants);
}
}

```

---

### 32. Print the Given Pattern

```

public class Pattern {
    public static void main(String[] args) {
        for (int i = 5; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("1 ");
            }
            System.out.println();
        }
    }
}

```

---

### 33. Feedback Collection System

```
import java.util.Scanner;

public class FeedbackSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int rating, count = 0, sum = 0;

        while (true) {
            System.out.println("Rate the product (1-5) or enter 0 to finish: ");
            rating = scanner.nextInt();

            if (rating == 0) {
                break;
            } else if (rating < 1 || rating > 5) {
                System.out.println("Invalid rating. Try again.");
                continue;
            }

            sum += rating;
            count++;
        }

        if (count > 0) {
            System.out.println("Average rating: " + (double) sum / count);
            System.out.println("Total ratings received: " + count);
        } else {
            System.out.println("No ratings received.");
        }
    }
}
```

---

### 34. Monthly Expenses Tracker

```
import java.util.Scanner;

public class MonthlyExpenses {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double total = 0;

        while (true) {
            System.out.println("Enter an expense (type 'done' to finish): ");
            String input = scanner.nextLine();

            if (input.equalsIgnoreCase("done")) {
                break;
            }

            try {
                total += Double.parseDouble(input);
            } catch (NumberFormatException e) {
                System.out.println("Invalid input. Please enter a valid number.");
            }
        }

        System.out.println("Total expenses for the month: $" + total);
    }
}
```

---

### 35. Password Validation System

```
import java.util.Scanner;

public class PasswordValidation {
```

```

public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    String password;

    while (true) {

        System.out.println("Create a password (at least 8 characters, 1 special character, 1 digit): ");

        password = scanner.nextLine();

        if (password.length() >= 8 && password.matches(".*\\W.*") && password.matches(".*\\d.*"))
        {

            System.out.println("Password created successfully.");

            break;

        } else {

            System.out.println("Password does not meet criteria. Try again.");

        }

    }

}

```

---

### 36. Fitness App for Logging Steps

```

import java.util.Scanner;

public class FitnessApp {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int totalSteps = 0;

        for (int i = 1; i <= 7; i++) {

            System.out.println("Enter steps for day " + i + ": ");

            totalSteps += scanner.nextInt();

        }

    }

}

```



```
        System.out.println("Total steps: " + totalSteps);

        System.out.println("Average steps per day: " + (totalSteps / 7.0));
    }
}
```

---

### 37. Temperature Conversion Tool

```
import java.util.Scanner;

public class TemperatureConverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Enter temperature value (type 'exit' to quit): ");
            String input = scanner.nextLine();

            if (input.equalsIgnoreCase("exit")) {
                break;
            }

            try {
                double temp = Double.parseDouble(input);
                System.out.println("Convert to (C)elsius or (F)ahrenheit? ");
                char unit = scanner.nextLine().toUpperCase().charAt(0);

                if (unit == 'C') {
                    System.out.println("Temperature in Celsius: " + ((temp - 32) * 5 / 9));
                } else if (unit == 'F') {
                    System.out.println("Temperature in Fahrenheit: " + ((temp * 9 / 5) + 32));
                } else {
```

```

        System.out.println("Invalid unit.");
    }
} catch (NumberFormatException e) {
    System.out.println("Invalid input. Please enter a number.");
}
}
}
}
}
}
}

```

Here's a response to each of the requested programs:

---

### 38. Banking System

```

import java.util.ArrayList;
import java.util.Scanner;

public class BankingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double balance = 0;
        ArrayList<String> transactionHistory = new ArrayList<>();

        while (true) {
            System.out.println("Choose an option: (1) Deposit (2) Withdraw (3) Exit");
            int choice = scanner.nextInt();

            if (choice == 1) {
                System.out.println("Enter deposit amount: ");
                double deposit = scanner.nextDouble();
                if (deposit > 0) {
                    balance += deposit;
                    transactionHistory.add("Deposited: $" + deposit);
                }
            }
        }
    }
}

```

```

        } else {
            System.out.println("Invalid deposit amount.");
        }
    } else if (choice == 2) {
        System.out.println("Enter withdrawal amount: ");
        double withdrawal = scanner.nextDouble();
        if (withdrawal > 0 && withdrawal <= balance) {
            balance -= withdrawal;
            transactionHistory.add("Withdrew: $" + withdrawal);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    } else if (choice == 3) {
        System.out.println("Exiting...");
        break;
    } else {
        System.out.println("Invalid choice. Try again.");
    }
}

System.out.println("Final balance: $" + balance);
System.out.println("Transaction History:");
for (String transaction : transactionHistory) {
    System.out.println(transaction);
}
}
}

```

---

### 39. Grade Input and Statistics

```
import java.util.Scanner;
```

```

public class GradeStatistics {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int totalGrades = 0, count = 0, passed = 0, highest = Integer.MIN_VALUE;

        System.out.println("Enter grades (enter -1 to stop):");

        while (true) {

            int grade = scanner.nextInt();

            if (grade == -1) break;

            totalGrades += grade;

            count++;

            if (grade > highest) highest = grade;

            if (grade >= 50) passed++;

        }

        if (count > 0) {

            double average = (double) totalGrades / count;

            System.out.println("Average Grade: " + average);

            System.out.println("Highest Grade: " + highest);

            System.out.println("Number of Students Passed: " + passed);

        } else {

            System.out.println("No grades entered.");

        }

    }

}

```

---

#### 40. Shopping Cart

```

import java.util.ArrayList;

import java.util.Scanner;

```

```

public class ShoppingCart {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        ArrayList<String> items = new ArrayList<>();

        ArrayList<Double> prices = new ArrayList<>();

        double total = 0;

        while (true) {

            System.out.println("Enter item name (type 'checkout' to finish): ");

            String item = scanner.nextLine();

            if (item.equalsIgnoreCase("checkout")) break;

            System.out.println("Enter price for " + item + ": ");

            double price = scanner.nextDouble();

            scanner.nextLine(); // consume newline

            items.add(item);

            prices.add(price);

            total += price;

        }

        System.out.println("Items Purchased:");

        for (int i = 0; i < items.size(); i++) {

            System.out.println(items.get(i) + ": $" + prices.get(i));

        }

        System.out.println("Total Amount Due: $" + total);

    }

}

```

---

#### 41. Total Sales and Commission

```

import java.util.Scanner;

```

```

public class SalesCommission {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double totalSales = 0;
        int count = 0;

        System.out.println("Enter sales figures (enter a negative number to stop):");
        while (true) {
            double sales = scanner.nextDouble();
            if (sales < 0) break;

            totalSales += sales;
            count++;
        }

        if (count > 0) {
            double averageSales = totalSales / count;
            System.out.println("Total Sales: $" + totalSales);
            System.out.println("Average Sales per Salesperson: $" + averageSales);
        } else {
            System.out.println("No sales entered.");
        }
    }
}

```

---

## 42. Reverse a String

```

import java.util.Scanner;

public class ReverseString {
    public static void main(String[] args) {

```

```
Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string:");

String input = scanner.nextLine();


String reversed = new StringBuilder(input).reverse().toString();

System.out.println("Reversed String: " + reversed);

}

}
```

---

#### **43. Check if String is a Palindrome**

```
import java.util.Scanner;


public class PalindromeCheck {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a string:");

        String input = scanner.nextLine();

        String reversed = new StringBuilder(input).reverse().toString();


        if (input.equalsIgnoreCase(reversed)) {

            System.out.println("The string is a palindrome.");

        } else {

            System.out.println("The string is not a palindrome.");

        }

    }

}
```

---

#### **44. Count Occurrences of Each Character**

```
import java.util.HashMap;

import java.util.Scanner;
```

```

public class CharacterOccurrences {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a string:");

        String input = scanner.nextLine();

        HashMap<Character, Integer> charCount = new HashMap<>();

        for (char c : input.toCharArray()) {

            charCount.put(c, charCount.getOrDefault(c, 0) + 1);

        }

        System.out.println("Character Occurrences:");

        charCount.forEach((key, value) -> System.out.println(key + ": " + value));

    }

}

```

---

#### **45. Reverse String Without reverse Method**

```

import java.util.Scanner;

public class ReverseWithoutMethod {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a string:");

        String input = scanner.nextLine();

        String reversed = "";

        for (int i = input.length() - 1; i >= 0; i--) {

            reversed += input.charAt(i);

        }

        System.out.println("Reversed String: " + reversed);

    }

}

```



