

1. Declare and initialize primitive types and print their default values

```
public class PrimitiveDefaults {
    byte byteVar;
    short shortVar;
    int intVar;
    long longVar;
    float floatVar;
    double doubleVar;
    char charVar;
    boolean booleanVar;

    public void printDefaults() {
        System.out.println("Default byte: " + byteVar);
        System.out.println("Default short: " + shortVar);
        System.out.println("Default int: " + intVar);
        System.out.println("Default long: " + longVar);
        System.out.println("Default float: " + floatVar);
        System.out.println("Default double: " + doubleVar);
        System.out.println("Default char: [" + charVar + "]");
        System.out.println("Default boolean: " + booleanVar);
    }

    public static void main(String[] args) {
        PrimitiveDefaults obj = new PrimitiveDefaults();
        obj.printDefaults();
    }
}
```

2. Detect overflow when adding two byte variables

```
public class ByteOverflowDetection {
    public static void main(String[] args) {
        byte a = 120;
        byte b = 10;

        int result = a + b;
        if (result > Byte.MAX_VALUE || result < Byte.MIN_VALUE) {
            System.out.println("Overflow detected! Result = " + result);
        } else {
            byte sum = (byte) result;
            System.out.println("No overflow. Sum = " + sum);
        }
    }
}
```

3. Type casting double to int and float to byte

```
public class TypeCastingExample {
    public static void main(String[] args) {
        double doubleValue = 123.456;
        float floatValue = 130.75f;

        int intValue = (int) doubleValue;
        byte byteValue = (byte) floatValue;

        System.out.println("Double to int: " + intValue);
        System.out.println("Float to byte: " + byteValue);
    }
}
```

4. Bitwise operations between int and byte

```
public class BitwiseOperations {
    public static void main(String[] args) {
        byte b = 0b0101;
        int i = 0b00110000;

        System.out.println("AND: " + (b & i));
        System.out.println("OR: " + (b | i));
        System.out.println("XOR: " + (b ^ i));
        System.out.println("NOT b: " + (~b));
    }
}
```

5. Accept input for all primitive types and display formatted

```
import java.util.Scanner;

public class PrimitiveInput {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        byte b = sc.nextByte();
        short s = sc.nextShort();
        int i = sc.nextInt();
        long l = sc.nextLong();
        float f = sc.nextFloat();
        double d = sc.nextDouble();
        char c = sc.next().charAt(0);
        boolean bool = sc.nextBoolean();

        System.out.printf("Byte: %d, Short: %d, Int: %d, Long: %d\n", b, s, i, l);
        System.out.printf("Float: %.2f, Double: %.3f\n", f, d);
    }
}
```

```
        System.out.println("Char: " + c + ", Boolean: " + bool);  
    }  
}
```