

Primitive Datatypes

1. Challenge: Declare and initialize variables of all primitive types and print their default values

Code:

```
public class PrimitiveDefaults {  
    public static void main(String[] args) {  
        byte b = 10;  
        short s = 100;  
        int i = 1000;  
        long l = 100000L;  
        float f = 10.5f;  
        double d = 20.99;  
        char c = 'A';  
        boolean bool = true;  
  
        System.out.println("byte: " + b);  
        System.out.println("short: " + s);  
        System.out.println("int: " + i);  
        System.out.println("long: " + l);  
        System.out.println("float: " + f);  
        System.out.println("double: " + d);  
        System.out.println("char: " + c);  
        System.out.println("boolean: " + bool);  
    }  
}
```

Output:

```
byte: 10  
short: 100  
int: 1000  
long: 100000  
float: 10.5  
double: 20.99  
char: A  
boolean: true
```

2. Challenge: Write a program to detect overflow when adding two byte variables.

Code:

```
public class ByteOverflow {  
    public static void main(String[] args) {  
        byte a = 127;  
        byte b = 1;  
        int result = a + b;  
  
        byte overflowedResult = (byte) result;  
  
        System.out.println("Actual sum: " + result); // Shows actual int value  
        System.out.println("Overflowed byte result: " + overflowedResult); //  
Shows overflowed byte value  
    }  
}
```

Output:

Actual sum: 128

Overflowed byte result: -128

3. Challenge: Use type casting to convert double to int and float to byte

Code:

```
public class TypeCasting {  
    public static void main(String[] args) {  
        double d = 10.75;  
        float f = 123.45f;  
  
        int intVal = (int) d; // fractional part lost  
        byte byteVal = (byte) f; // precision lost  
  
        System.out.println("double to int: " + intVal);  
        System.out.println("float to byte: " + byteVal);  
    }  
}
```

Output:

double to int: 10

float to byte: 123

4. Challenge: Perform bitwise operations between int and byte

Code:

```
public class BitwiseDemo {  
    public static void main(String[] args) {  
        int a = 12;    // binary: 00001100  
        byte b = 5;    // binary: 00000101  
  
        System.out.println("a & b = " + (a & b)); // AND  
        System.out.println("a | b = " + (a | b)); // OR  
        System.out.println("a ^ b = " + (a ^ b)); // XOR  
        System.out.println("~a = " + (~a));      // NOT  
    }  
}
```

Output:

a & b = 4

a | b = 13

a ^ b = 9

~a = -13

5. Challenge: Accept input for all primitive types and display them formatted

Code:

```
import java.util.Scanner;  
  
public class AllPrimitiveInput {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter byte: ");  
        byte b = sc.nextByte();  
  
        System.out.print("Enter short: ");  
        short s = sc.nextShort();  
  
        System.out.print("Enter int: ");  
        int i = sc.nextInt();  
  
        System.out.print("Enter long: ");  
        long l = sc.nextLong();  
    }  
}
```

```

    System.out.print("Enter float: ");
    float f = sc.nextFloat();

    System.out.print("Enter double: ");
    double d = sc.nextDouble();

    System.out.print("Enter char: ");
    char c = sc.next().charAt(0);

    System.out.print("Enter boolean: ");
    boolean bool = sc.nextBoolean();

    System.out.println("\n--- Formatted Output ---");
    System.out.printf("Byte: %d\nShort: %d\nInt: %d\nLong: %d\n", b, s,
i, l);
    System.out.printf("Float: %.2f\nDouble: %.2f\n", f, d);
    System.out.println("Char: " + c);
    System.out.println("Boolean: " + bool);

    sc.close();
}
}

```

Output:

```

Enter byte: 12
Enter short: 1000
Enter int: 12345
Enter long: 999999
Enter float: 1.23
Enter double: 3.14159
Enter char: Z
Enter boolean: true

```