**ASSIGNMENT NO 2**

**1. Working with java. lang. Boolean:**

**a. Exploring java. lang. Boolean:**

* **Modifiers: public final class Boolean extends Object implements Comparable<Boolean>, Serializable**
* **Super types: Object, Comparable<Boolean>, Serializable**

**b. Converting a boolean to a string:**

public class BooleanEx {

public static void main (String[] args) {

boolean st = true;

String str= Boolean.toString(st);

System.out.println("Boolean value as a string: " + str);

}

}

**c. Converting a string to a boolean:**

public class BooleanExample {

public static void main(String[] args) {

String str = "true";

boolean st = Boolean.parseBoolean(str);

System.out.println("Boolean value from string: " + st);

}

}

**d. Handling a NumberFormatException:**

public class BooleanExample {

public static void main(String[] args) {

String str= "1";

try {

boolean st = Boolean.parseBoolean(str);

System.out.println("Boolean value from string: " + st);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for boolean: " + e.getMessage());

}

}

}

**e. Converting a boolean to a Boolean wrapper:**

public class BooleanExample {

public static void main(String[] args) {

boolean st= true;

Boolean wrapper = Boolean.valueOf(st);

System.out.println("Boolean wrapper: " + wrapper);

}

}

**f. Converting a string to a Boolean wrapper:**

public class BooleanExample {

public static void main(String[] args) {

String strs = "true";

Boolean wrapper = Boolean.valueOf(str);

System.out.println("Boolean wrapper: " + wrapper);

}

}

**g. Converting a boolean to other primitive types and vice versa:**

**Note: Converting a boolean to other primitive types or vice versa is not directly supported in Java. However, you can achieve similar results using conditional logic and casting:**

public class BooleanExample {

public static void main(String[] args) {

boolean st = true;

// Convert boolean to int (1 for true, 0 for false)

int intValue = st ? 1 : 0;

// Convert boolean to long (1 for true, 0 for false)

long longValue =st? 1 : 0;

// Convert boolean to float (1.0f for true, 0.0f for false)

float floatValue =st? 1.0f : 0.0f;

// Convert boolean to double (1.0d for true, 0.0d for false)

double doubleValue =st? 1.0d : 0.0d;

// Convert int, long, float, or double to boolean

// (true if non-zero, false if zero)

boolean booleanFromInt = intValue != 0;

boolean booleanFromLong = longValue != 0;

boolean booleanFromFloat = floatValue != 0.0f;

boolean booleanFromDouble = doubleValue != 0.0d;

System.out.println("Boolean to int: " + intValue);

System.out.println("Boolean to long: " + longValue);

System.out.println("Boolean to float: " + floatValue);

System.out.println("Boolean to double: " + doubleValue);

System.out.println("Int to boolean: " + booleanFromInt);

System.out.println("Long to boolean: " + booleanFromLong);

System.out.println("Float to boolean: " + booleanFromFloat);

System.out.println("Double to boolean: " + booleanFromDouble);

}

}

===========================================================================

**2 . Working with java.lang.Byte:**

1. **Exploring java.lang.Byte:**

* **Modifiers:** public final class Byte extends Number implements Comparable<Byte>
* **Super types:** Number, Comparable<Byte>

**b. Testing the number of bytes used to represent a byte value:**

public class ByteEx {

public static void main(String[] args) {

System.out.println("Number of bytes used to represent a byte value: " + Byte.BYTES);

}

}

**c. Finding the minimum and maximum values of a byte:**

public class ByteExample {

public static void main(String[] args) {

System.out.println("Minimum byte value: " + Byte.MIN\_VALUE);

System.out.println("Maximum byte value: " + Byte.MAX\_VALUE);

}

}

**d. Converting a byte to a string:**

public class ByteExample {

public static void main(String[] args) {

byte number = 42;

String strNumber = Byte.toString(number);

System.out.println("Byte value as a string: " + strNumber);

}

}

**e. Converting a string to a byte:**

public class ByteExample {

public static void main(String[] args) {

String strNumber = "123";

byte number = Byte.parseByte(strNumber);

System.out.println("Byte value from string: " + number);

}

}

**f. Handling a NumberFormatException:**

public class ByteExample {

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

try {

byte number = Byte.parseByte(strNumber);

System.out.println("Byte value from string: " + number);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for byte: " + e.getMessage());

}

}

}

**g. Converting a byte to a Byte wrapper:**

public class ByteExample {

public static void main(String[] args) {

byte number = 42;

Byte wrapper = Byte.valueOf(number);

System.out.println("Byte wrapper: " + wrapper);

}

}

**h. Converting a string to a Byte wrapper:**

public class ByteExample {

public static void main(String[] args) {

String strNumber = "123";

Byte wrapper = Byte.valueOf(strNumber);

System.out.println("Byte wrapper: " + wrapper);

}

}

**i. Converting a byte to other primitive types and vice versa:**

public class ByteExample {

public static void main(String[] args) {

byte number = 42;

int intValue = number;

long longValue = number;

float floatValue = number;

double doubleValue = number;

System.out.println("Byte to int: " + intValue);

System.out.println("Byte to long: " + longValue);

System.out.println("Byte to float: " + floatValue);

System.out.println("Byte to double: " + doubleValue);

// Conversion from other primitive types to byte

// Note: Potential loss of precision or range issues

byte byteValue = (byte) intValue;

System.out.println("Int to byte: " + byteValue);

}

}

=========================================================================

**3. Working with** **java.lang.Short:**

**a. Exploring java.lang.Short:**

* **Modifiers:** public final class Short extends Number implements Comparable<Short>
* **Super types:** Number, Comparable<Short>

**b. Testing the number of bytes used to represent a short value:**

public class ShortExample {

public static void main(String[] args) {

System.out.println("Number of bytes used to represent a short value: " + Short.BYTES);

}

}

**c. Finding the minimum and maximum values of a short:**

public class ShortExample {

public static void main(String[] args) {

System.out.println("Minimum short value: " + Short.MIN\_VALUE);

System.out.println("Maximum short value: " + Short.MAX\_VALUE);

}

}

**d. Converting a short to a string:**

public class ShortExample {

public static void main(String[] args) {

short number = 42;

String strNumber = Short.toString(number);

System.out.println("Short value as a string: " + strNumber);

}

}

**e. Converting a string to a short:**

public class ShortExample {

public static void main(String[] args) {

String strNumber = "123";

short number = Short.parseShort(strNumber);

System.out.println("Short value from string: " + number);

}

}

**f. Handling a NumberFormatException:**

public class ShortExample {

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

try {

short number = Short.parseShort(strNumber);

System.out.println("Short value from string: " + number);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for short: " + e.getMessage());

}

}

}

**g. Converting a short to a Short wrapper:**

public class ShortExample {

public static void main(String[] args) {

short number = 42;

Short wrapper = Short.valueOf(number);

System.out.println("Short wrapper: " + wrapper);

}

}

**h. Converting a string to a Short wrapper:**

public class ShortExample {

public static void main(String[] args) {

String strNumber = "123";

Short wrapper = Short.valueOf(strNumber);

System.out.println("Short wrapper: " + wrapper);

}

}

**i. Converting a short to other primitive types and vice versa:**

public class ShortExample {

public static void main(String[] args) {

short number = 42;

int intValue = number;

long longValue = number;

float floatValue = number;

double doubleValue = number;

System.out.println("Short to int: " + intValue);

System.out.println("Short to long: " + longValue);

System.out.println("Short to float: " + floatValue);

System.out.println("Short to double: " + doubleValue);

// Conversion from other primitive types to short

// Note: Potential loss of precision or range issues

short shortValue = (short) intValue;

System.out.println("Int to short: " + shortValue);

}

}

===========================================================================

**4. Working with java.lang.Integer:**

**a. Exploring java.lang.Integer:**

* **Modifiers:** public final class Integer extends Number implements Comparable<Integer>
* **Super types:** Number, Comparable<Integer>

**b. Testing the number of bytes used to represent an int value:**

public class IntegerExample {

public static void main(String[] args) {

System.out.println("Number of bytes used to represent an int value: " + Integer.BYTES);

}

}

**c. Finding the minimum and maximum values of an int:**

public class IntegerExample {

public static void main(String[] args) {

System.out.println("Minimum int value: " + Integer.MIN\_VALUE);

System.out.println("Maximum int value: " + Integer.MAX\_VALUE);

}

}

**d. Converting an int to a string:**

public class IntegerExample {

public static void main(String[] args) {

int number = 42;

String strNumber = Integer.toString(number);

System.out.println("Int value as a string: " + strNumber);

}

}

**e. Converting a string to an int:**

public class IntegerExample {

public static void main(String[] args) {

String strNumber = "123";

int number = Integer.parseInt(strNumber);

System.out.println("Int value from string: " + number);

}

}

**f. Handling a NumberFormatException:**

public class IntegerExample {

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

try {

int number = Integer.parseInt(strNumber);

System.out.println("Int value from string: " + number);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for int: " + e.getMessage());

}

}

}

**g. Converting an int to an Integer wrapper:**

public class IntegerExample {

public static void main(String[] args) {

int number = 42;

Integer wrapper = Integer.valueOf(number);

System.out.println("Integer wrapper: " + wrapper);

}

}

**h. Converting a string to an Integer wrapper:**

public class IntegerExample {

public static void main(String[] args) {

String strNumber = "123";

Integer wrapper = Integer.valueOf(strNumber);

System.out.println("Integer wrapper: " + wrapper);

}

}

**i. Adding two integers using Integer. Sum():**

public class IntegerExample {

public static void main(String[] args) {

int num1 = 10;

int num2 = 20;

int sum = Integer.sum(num1, num2);

System.out.println("Sum of " + num1 + " and " + num2 + " is: " + sum);

}

}

**j. Finding the minimum and maximum values using Integer.min() and Integer.max():**

public class IntegerExample {

public static void main(String[] args) {

int num1 = 10;

int num2 = 20;

int min = Integer.min(num1, num2);

int max = Integer.max(num1, num2);

System.out.println("Minimum of " + num1 + " and " + num2 + " is: " + min);

System.out.println("Maximum of " + num1 + " and " + num2 + " is: " + max);

}

}

**k. Converting an int to binary, octal, and hexadecimal:**

public class IntegerExample {

public static void main(String[] args) {

int number = 7;

String binaryString = Integer.toBinaryString(number);

String octalString = Integer.toOctalString(number);

String hexadecimalString = Integer.toHexString(number);

System.out.println("Binary representation: " + binaryString);

System.out.println("Octal representation: " + octalString);

System.out.println("Hexadecimal representation: " + hexadecimalString);

}

}

**l. Converting an int to other primitive types and vice versa:**

public class IntegerExample {

public static void main(String[] args) {

int number = 42;

byte byteValue = (byte) number;

short shortValue = (short) number;

long longValue = number;

float floatValue = number;

double doubleValue = number;

System.out.println("Int to byte: " + byteValue);

System.out.println("Int to short: " + shortValue);

System.out.println("Int to long: " + longValue);

System.out.println("Int to float: " + floatValue);

System.out.println("Int to double: " + doubleValue);

// Conversion from other primitive types to int

// Note: Potential loss of precision or range issues

int intValue = (int) longValue;

System.out.println("Long to int: " + intValue);

}

}

===========================================================================

**5. Working with java.lang.Long:**

**a. Exploring java.lang.Long:**

* **Modifiers:** public final class Long extends Number implements Comparable<Long>
* **Super types:** Number, Comparable<Long>

**b. Testing the number of bytes used to represent a long value:**

public class LongExample {

public static void main(String[] args) {

System.out.println("Number of bytes used to represent a long value: " + Long.BYTES);

}

}

**c. Finding the minimum and maximum values of a long:**

public class LongExample {

public static void main(String[] args) {

System.out.println("Minimum long value: " + Long.MIN\_VALUE);

System.out.println("Maximum long value: " + Long.MAX\_VALUE);

}

}

**d. Converting a long to a string:**

public class LongExample {

public static void main(String[] args) {

long number = 42;

String strNumber = Long.toString(number);

System.out.println("Long value as a string: " + strNumber);

}

}

**e. Converting a string to a long:**

public class LongExample {

public static void main(String[] args) {

String strNumber = "123";

long number = Long.parseLong(strNumber);

System.out.println("Long value from string: " + number);

}

}

**f. Handling a NumberFormatException:**

public class LongExample {

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

try {

long number = Long.parseLong(strNumber);

System.out.println("Long value from string: " + number);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for long: " + e.getMessage());

}

}

}

**g. Converting a long to a Long wrapper:**

public class LongExample {

public static void main(String[] args) {

long number = 42;

Long wrapper = Long.valueOf(number);

System.out.println("Long wrapper: " + wrapper);

}

}

**h. Converting a string to a Long wrapper:**

public class LongExample {

public static void main(String[] args) {

String strNumber = "123";

Long wrapper = Long.valueOf(strNumber);

System.out.println("Long wrapper: " + wrapper);

}

}

**i. Adding two longs using Long.sum():**

public class LongExample {

public static void main(String[] args) {

long num1 = 1123;

long num2 = 9845;

long sum = Long.sum(num1, num2);

System.out.println("Sum of " + num1 + " and " + num2 + " is: " + sum);

}

}

**j. Finding the minimum and maximum values using Long.min() and Long.max():**

public class LongExample {

public static void main(String[] args) {

long num1 = 1122;

long num2 = 5566;

long min = Long.min(num1, num2);

long max = Long.max(num1, num2);

System.out.println("Minimum of " + num1 + " and " + num2 + " is: " + min);

System.out.println("Maximum of " + num1 + " and " + num2 + " is: " + max);

}

}

**k. Converting a long to binary, octal, and hexadecimal:**

public class LongExample {

public static void main(String[] args) {

long number = 7;

String binaryString = Long.toBinaryString(number);

String octalString = Long.toOctalString(number);

String hexadecimalString = Long.toHexString(number);

System.out.println("Binary representation: " + binaryString);

System.out.println("Octal representation: " + octalString);

System.out.println("Hexadecimal representation: " + hexadecimalString);

}

}

**l. Converting a long to other primitive types and vice versa:**

public class LongExample {

public static void main(String[] args) {

long number = 42;

byte byteValue = (byte) number;

short shortValue = (short) number;

int intValue = (int) number;

float floatValue = number;

double doubleValue = number;

System.out.println("Long to byte: " + byteValue);

System.out.println("Long to short: " + shortValue);

System.out.println("Long to int: " + intValue);

System.out.println("Long to float: " + floatValue);

System.out.println("Long to double: " + doubleValue);

// Conversion from other primitive types to long

// Note: Potential loss of precision or range issues

long longValue = (long) intValue;

System.out.println("Int to long: " + longValue);

}

}

===========================================================================

**6. Working with java.lang.Float:**

**a. Exploring java.lang.Float:**

* **Modifiers:** public final class Float extends Number implements Comparable<Float>
* **Super types:** Number, Comparable<Float>

**b. Testing the number of bytes used to represent a float value:**

Java

public class FloatExample {

public static void main(String[] args) {

System.out.println("Number of bytes used to represent a float value: " + Float.BYTES);

}

}

**c. Finding the minimum and maximum values of a float:**

Java

public class FloatExample {

public static void main(String[] args) {

System.out.println("Minimum float value: " + Float.MIN\_VALUE);

System.out.println("Maximum float value: " + Float.MAX\_VALUE);

}

}

**d. Converting a float to a string:**

Java

public class FloatExample {

public static void main(String[] args) {

float number = 42.0f;

String strNumber = Float.toString(number);

System.out.println("Float value as a string: " + strNumber);

}

}

**e. Converting a string to a float:**

Java

public class FloatExample {

public static void main(String[] args) {

String strNumber = "123.45";

float number = Float.parseFloat(strNumber);

System.out.println("Float value from string: " + number);

}

}

**f. Handling a NumberFormatException:**

Java

public class FloatExample {

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

try {

float number = Float.parseFloat(strNumber);

System.out.println("Float value from string: " + number);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for float: " + e.getMessage());

}

}

}

**g. Converting a float to a Float wrapper:**

Java

public class FloatExample {

public static void main(String[] args) {

float number = 42.0f;

Float wrapper = Float.valueOf(number);

System.out.println("Float wrapper: " + wrapper);

}

}

**h. Converting a string to a Float wrapper:**

Java

public class FloatExample {

public static void main(String[] args) {

String strNumber = "123.45";

Float wrapper = Float.valueOf(strNumber);

System.out.println("Float wrapper: " + wrapper);

}

}

**i. Adding two floats using Float.sum():**

Java

public class FloatExample {

public static void main(String[] args) {

float num1 = 112.3f;

float num2 = 984.5f;

float sum = Float.sum(num1, num2);

System.out.println("Sum of " + num1 + " and " + num2 + " is: " + sum);

}

}

**j. Finding the minimum and maximum values using Float.min() and Float.max():**

Java

public class FloatExample {

public static void main(String[] args) {

float num1 = 112.2f;

float num2 = 556.6f;

float min = Float.min(num1, num2);

float max = Float.max(num1, num2);

System.out.println("Minimum of " + num1 + " and " + num2 + " is: " + min);

System.out.println("Maximum of " + num1 + " and " + num2 + " is: " + max);

}

}

**k. Finding the square root of a float using Math.sqrt():**

Java

public class FloatExample {

public static void main(String[] args) {

float number = -25.0f;

float sqrt = (float) Math.sqrt(number);

System.out.println("Square root of " + number + " is: " + sqrt);

}

}

**l. Dividing two floats with the same value:**

Java

public class FloatExample {

public static void main(String[] args) {

float num1 = 0.0f;

float num2 = 0.0f;

float result = num1 / num2;

System.out.println("Dividing " + num1 + " by " + num2 + " is: " + result);

}

}

**m. Converting a float to other primitive types and vice versa:**

public class FloatExample {

public static void main(String[] args) {

float number = 42.0f;

int intValue = (int) number;

long longValue = (long) number;

byte byteValue = (byte) number;

short shortValue = (short) number;

double doubleValue = number;

System.out.println("Float to int: " + intValue);

System.out.println("Float to long: " + longValue);

System.out.println("Float to byte: " + byteValue);

System.out.println("Float to short: " + shortValue);

System.out.println("Float to double: " + doubleValue);

// Conversion from other primitive types to float

// Note: Potential loss of precision or range issues

float floatValue = (float) intValue;

System.out.println("Int to float: " + floatValue);

}

}

===========================================================================

**7. Working with java.lang.Double:**

**a. Exploring java.lang.Double:**

* **Modifiers:** public final class Double extends Number implements Comparable<Double>
* **Super types:** Number, Comparable<Double>

**b. Testing the number of bytes used to represent a double value:**public class DoubleExample {

public static void main(String[] args) {

System.out.println("Number of bytes used to represent a double value: " + Double.BYTES);

}

}

**c. Finding the minimum and maximum values of a double:**

public class DoubleExample {

public static void main(String[] args) {

System.out.println("Minimum double value: " + Double.MIN\_VALUE);

System.out.println("Maximum double value: " + Double.MAX\_VALUE);

}

}

**d. Converting a double to a string:**

public class DoubleExample {

public static void main(String[] args) {

double number = 42.0;

String strNumber = Double.toString(number);

System.out.println("Double value as a string: " + strNumber);

}

}

**e. Converting a string to a double:**

public class DoubleExample {

public static void main(String[] args) {

String strNumber = "123.45";

double number = Double.parseDouble(strNumber);

System.out.println("Double value from string: " + number);

}

}

**f. Handling a NumberFormatException:**

public class DoubleExample {

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

try {

double number = Double.parseDouble(strNumber);

System.out.println("Double value from string: " + number);

} catch (NumberFormatException e) {

System.out.println("Invalid string format for double: " + e.getMessage());

}

}

}

**g. Converting a double to a Double wrapper:**

public class DoubleExample {

public static void main(String[] args) {

double number = 42.0;

Double wrapper = Double.valueOf(number);

System.out.println("Double wrapper: " + wrapper);

}

}

**h. Converting a string to a Double wrapper:**

public class DoubleExample {

public static void main(String[] args) {

String strNumber = "123.45";

Double wrapper = Double.valueOf(strNumber);

System.out.println("Double wrapper: " + wrapper);

}

}

**i. Adding two doubles using Double.sum():**

public class DoubleExample {

public static void main(String[] args) {

double num1 = 112.3;

double num2 = 984.5;

double sum = Double.sum(num1, num2);

System.out.println("Sum of " + num1 + " and " + num2 + " is: " + sum);

}

}

**j. Finding the minimum and maximum values using Double.min() and Double.max():**

public class DoubleExample {

public static void main(String[] args) {

double num1 = 112.2;

double num2 = 556.6;

double min = Double.min(num1, num2);

double max = Double.max(num1, num2);

System.out.println("Minimum of " + num1 + " and " + num2 + " is: " + min);

System.out.println("Maximum of " + num1 + " and " + num2 + " is: " + max);

}

}

**k. Finding the square root of a double using Math.sqrt():**

public class DoubleExample {

public static void main(String[] args) {

double number = -25.0;

double sqrt = Math.sqrt(number);

System.out.println("Square root of " + number + " is: " + sqrt);

}

}

**l. Dividing two doubles with the same value:**

public class DoubleExample {

public static void main(String[] args) {

double num1 = 0.0;

double num2 = 0.0;

double result = num1 / num2;

System.out.println("Dividing " + num1 + " by " + num2 + " is: " + result);

}

}

**m. Converting a double to other primitive types and vice versa:**

public class DoubleExample {

public static void main(String[] args) {

double number = 42.0;

int intValue = (int) number;

long longValue = (long) number;

float floatValue = (float) number;

byte byteValue = (byte) number;

short shortValue = (short) number;

System.out.println("Double to int: " + intValue);

System.out.println("Double to long: " + longValue);

System.out.println("Double to float: " + floatValue);

System.out.println("Double to byte: " + byteValue);

System.out.println("Double to short: " + shortValue);

// Conversion from other primitive types to double

// Note: Potential loss of precision or range issues

double doubleValue = (double) intValue;

System.out.println("Int to double: " + doubleValue);

}

}

===========================================================================

**8. Conversion between Primitive Types and Strings:**

public class ConversionExample {

public static void main(String[] args) {

byte byteValue = 42;

short shortValue = 123;

int intValue = 456;

long longValue = 7890L;

float floatValue = 12.34f;

double doubleValue = 56.78;

boolean booleanValue = true;

char charValue = 'A';

// Using to String method of wrapper classes

String byteString1 = Byte.toString(byteValue);

String shortString1 = Short.toString(shortValue);

String intString1 = Integer.toString(intValue);

String longString1 = Long.toString(longValue);

String floatString1 = Float.toString(floatValue);

String doubleString1 = Double.toString(doubleValue);

String booleanString1 = Boolean.toString(booleanValue);

String charString1 = Character.toString(charValue);

// Using value Of method of String class

String byteString2 = String.valueOf(byteValue);

String shortString2 = String.valueOf(shortValue);

String intString2 = String.valueOf(intValue);

String longString2 = String.valueOf(longValue);

String floatString2 = String.valueOf(floatValue);

String doubleString2 = String.valueOf(doubleValue);

String booleanString2 = String.valueOf(booleanValue);

String charString2 = String.valueOf(charValue);

// Print the results

System.out.println("Using toString method:");

System.out.println("byte: " + byteString1);

System.out.println("short: " + shortString1);

System.out.println("int: " + intString1);

System.out.println("long: " + longString1);

System.out.println("float: " + floatString1);

System.out.println("double: " + doubleString1);

System.out.println("boolean: " + booleanString1);

System.out.println("char: " + charString1);

System.out.println("\nUsing valueOf method:");

System.out.println("byte: " + byteString2);

System.out.println("short: " + shortString2);

System.out.println("int: " + intString2);

System.out.println("long: " + longString2);

System.out.println("float: " + floatString2);

System.out.println("double: " + doubleString2);

System.out.println("boolean: " + booleanString2);

System.out.println("char: " + charString2);

}

}

===========================================================================

**9. Default Values of Primitive Types:**

public class DefaultValuesExample {

byte byteField;

short shortField;

int intField;

long longField;

float floatField;

double doubleField;

boolean booleanField;

char charField;

public static void main(String[] args) {

DefaultValuesExample obj = new DefaultValuesExample();

System.out.println("Default values of instance variables:");

System.out.println("byte: " + obj.byteField);

System.out.println("short: " + obj.shortField);

System.out.println("int: " + obj.intField);

System.out.println("long: " + obj.longField);

System.out.println("float: " + obj.floatField);

System.out.println("double: " + obj.doubleField);

System.out.println("boolean: " + obj.booleanField);

System.out.println("char: " + obj.charField);

// Default values of static variables

System.out.println("\n Default values of static variables:");

System.out.println("byte: " + byteField);

System.out.println("short: " + shortField);

System.out.println("int: " + intField);

System.out.println("long: " + longField);

System.out.println("float:   " + floatField);

System.out.println("double: " + doubleField);

System.out.println("boolean:" + booleanField);

System.out.println("char: " + charField);

}

}

===========================================================================

**10. Arithmetic Operations with Command Line Input:**

public class ArithmeticOperations {

public static void main(String[] args) {

if (args.length != 3) {

System.out.println("Usage: java ArithmeticOperations <operand1> <operator> <operand2>");

return;

}

int operand1 = Integer.parseInt(args[0]);

char operator = args[1].charAt(0);

int operand2 = Integer.parseInt(args[2]);

int result = 0;

switch (operator) {

case '+':

result = operand1 + operand2;

break;

case '-':

result = operand1 - operand2;

break;

case  '\*':

result = operand1 \* operand2;

break;

case '/':

result = operand1 / operand2;

break;

default:

System.out.println("Invalid operator: " + operator);

return;

}

System.out.println("Result: " + result);

}

}

