1. **Working With Java.lang.boolean**

**b. Declare a method-local variable status of type boolean with the value true and convert it to a String using the toString method. (Hint: Use Boolean.toString(Boolean) ).**

public class Main {

public static void main(String [] args){

boolean status = true;

String str = Boolean.toString(status);

System.out.println(str);

}

}

o/p = true

**c. Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method. (Hint: Use Boolean.parseBoolean(String)).**

public class Main {

public static void main(String [] args){

String strStatus = "true";

boolean str = Boolean.parseBoolean(strStatus);

System.out.println(str);

}

}

o/p = true

**d. Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean. (Hint: parseBoolean method will not work as expected with "1" or "0").**

public class Main {

public static void main(String [] args){

String strStatus = "1";

boolean str = "1".equals(strStatus);

System.out.println(str);

}

}

o\p = true ; if we use parseBoolean it will give false for both 1 and 0.

**e. Declare a method-local variable status of type boolean with the value true and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(boolean)).**

public class Main

{

public static void main(String[] args) {

boolean status= true;

Boolean statusStr = Boolean.valueOf(status);

System.out.println(statusStr);

}

}

o/p= true

**f. Declare a method-local variable strStatus of type String with the value "true" and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(String)).**

public class Main

{

public static void main(String[] args) {

String status= "true";

Boolean statusStr = Boolean.valueOf(status);

System.out.println(statusStr);

}

}

**G .Experiment with converting a boolean value into other primitive types or vice versa and observe the results.**

1. **Working with java.lang.Byte**

**b. Write a program to test how many bytes are used to represent a byte value using the BYTES field. (Hint: Use Byte.BYTES**

public class Main

{

public static void main(String[] args) {

Byte a = 10;

byte b = a.byteValue();

System.out.println(b);

}

}

o/p = 10;

**c. Write a program to find the minimum and maximum values of byte using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Byte.MIN\_VALUE and Byte.MAX\_VALUE).**

public class Main

{

public static void main(String[] args) {

byte minValue = Byte.MIN\_VALUE;

byte maxValue = Byte.MAX\_VALUE;

System.out.println(minValue);

System.out.println(maxValue);

}

}

o/p = -128 127

**d. Declare a method-local variable number of type byte with some value and convert it to a String using the toString method. (Hint: Use Byte.toString(byte)).**

public class Main

{

public static void main(String[] args) {

byte number = 15;

String str = Byte.toString(number);

System.out.println(str);

}

}

o/p = 15;

**e. Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method. (Hint: Use Byte.parseByte(String)).**

public class Main

{

public static void main(String[] args) {

String str = “15”

byte byatevalue = Byte.parseByte(str);

System.out.println(byatevalue);

}

}

o/p =15

**`f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value. (Hint: parseByte method will throw a NumberFormatException).**

public class Main {

public static void main(String[] args) {

value "Ab12Cd3"

String strNumber = "Ab12Cd3";

NumberFormatException)

byte byteValue = Byte.parseByte(strNumber);

program

System.out.println("Converted value: " + byteValue);

}

}

o/p = exception

we can do it using exception handling.

**g. Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(byte)).**

public class Main

{

public static void main(String[] args) {

byte value = 42;

byte byatevalue = Byte.valueOf(value);

System.out.println(byatevalue);

}

}

o/p =42;

**h. Declare a method-local variable strNumber of type String with some byte value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "42";

byte byatevalue = Byte.valueOf(strNumber);

System.out.println(byatevalue);

}

}

o/p = 42

**i. Experiment with converting a byte value into other primitive types or vice versa and observe the results.**

public class Main {

public static void main(String[] args) {

byte myByte = 42;

int myInt = myByte;

System.out.println(myInt);

long myLong = myByte;

System.out.println(myLong);

double myDouble = myByte;

System.out.println(myDouble);

char myChar = (char) myByte;

System.out.println(myChar);

}

}

o/p = 42

42

42.0

\*

public class Main {

public static void main(String[] args) {

int myInt = 123;

byte intToByte = (byte) myInt;

System.out.println(intToByte);

long myLong = 9876543210L;

byte longToByte = (byte) myLong;

System.out.println(longToByte);

double myDouble = 3.14;

byte doubleToByte = (byte) myDouble;

System.out.println(doubleToByte);

}

}

o/p = 123

-22

3

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

**b. Write a program to test how many bytes are used to represent a short value using the BYTES field. (Hint: Use Short.BYTES).**

public class Main{

public static void main(String [] args){

int sizeInByte = Short.BYTES;

System.out.println(sizeInByte);

}

}

o/p = 0

**c. Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Short.MIN\_VALUE and Short.MAX\_VALUE).**

public class Main{

public static void main(String [] args){

int max =Short.MAX\_VALUE;

int min =Short.MIN\_VALUE;

System.out.println(min +" "+max);

}

}

o/p = -32768 32767

**d. Declare a method-local variable number of type short with some value and convert it to a String using the toString method. (Hint: Use Short.toString(short)).**

public class Main{

public static void main(String [] args){

short number = 1;

String str = Short.toString(number);

System.out.println(str);

}

}

o/p = 1

**e. Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method. (Hint: Use Short.parseShort(String)).**

public class Main{

public static void main(String [] args){

String strNumber = "123";

short str = Short.parseShort(strNumber);

System.out.println(str);

}

}

o/p = 123

**f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value. (Hint: parseShort method will throw a NumberFormatException).**

public class Main{

public static void main(String [] args){

String strNumber = "Ab12Cd3";

short str = Short.parseShort(strNumber);

System.out.println(str);

}

}

**Exception in** thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"

at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.base/java.lang.Integer.parseInt(Integer.java:652)

at java.base/java.lang.Short.parseShort(Short.java:120)

at java.base/java.lang.Short.parseShort(Short.java:146)

at Main.main(Main.java:5)

**g. Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(short)).**

public class Main{

public static void main(String [] args){

short strNumber = 10;

short str = Short.valueOf(strNumber);

System.out.println(str);

}

}

o/p = 10

**h. Declare a method-local variable strNumber of type String with some short value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(String)).**

public class Main{

public static void main(String [] args){

String strNumber = "3000";

short str = Short.valueOf(strNumber);

System.out.println(str);

}

}

**i. Experiment with converting a short value into other primitive types or vice versa and observe the results.**

public class Main {

public static void main(String[] args) {

short myShort = 12345;

int myInt = myShort;

System.out.println( myInt);

long myLong = myShort;

System.out.println( myLong);

double myDouble = myShort;

System.out.println(myDouble);

char myChar = (char) myShort;

System.out.println(myChar);

}

}

o/p = 12345

12345

12345.0

〹

public class Main {

public static void main(String[] args) {

int myInt = 56789;

short intToShort = (short) myInt;

System.out.println(intToShort);

long myLong = 9876543210L;

short longToShort = (short) myLong;

System.out.println(longToShort);

double myDouble = 123.456;

short doubleToShort = (short) myDouble;

System.out.println(doubleToShort);

}

}

o/p = -8747

5866

123

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

**b. Write a program to test how many bytes are used to represent an int value using the BYTES field. (Hint: Use Integer.BYTES).**

public class Main

{

public static void main(String[] args) {

int intSizeInBytes = Integer.BYTES;

System.out.println(intSizeInBytes);

}

}

o/p = 4;

**c. Write a program to find the minimum and maximum values of int using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Integer.MIN\_VALUE and Integer.MAX\_VALUE).**

public class Main

{

public static void main(String[] args) {

int minValue = Integer.MIN\_VALUE;

int maxValue = Integer.MAX\_VALUE;

System.out.println( minValue);

System.out.println( maxValue);

}

}

o/p = -2147483648

2147483647

**d. Declare a method-local variable number of type int with some value and convert it to a String using the toString method. (Hint: Use Integer.toString(int)).**

public class Main

{

public static void main(String[] args) {

int number = 10;

String str = Integer.toString(number);

System.out.println(str);

}

}

o/p = 10

**e. Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method. (Hint: Use Integer.parseInt(String)).**

public class Main

{

public static void main(String[] args) {

String str = "10";

int num = Integer.parseInt(str);

System.out.println(num);

}

}

o/p = 10

**f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to an int value. (Hint: parseInt method will throw a NumberFormatException).**

public class Main

{

public static void main(String[] args) {

String str = "10abc";

int num = Integer.parseInt(str);

System.out.println(num);

}

}

o/p = Exception in thread "main" java.lang.NumberFormatException: For input string: "10abc"

at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.base/java.lang.Integer.parseInt(Integer.java:652)

at java.base/java.lang.Integer.parseInt(Integer.java:770)

at Main.main(Main.java:13)

**g. Declare a method-local variable number of type int with some value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(int)).**

public class Main

{

public static void main(String[] args) {

int num = 10;

int num2 = Integer.valueOf(num);

System.out.println(num2);

}

}

o/p = 10;

**h. Declare a method-local variable strNumber of type String with some integer value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(String)).**

public class Main

{

public static void main(String[] args) {

String str = "123";

int num = Integer.valueOf(str);

System.out.println(num);

}

}

o/p = 123;

**i. Declare two integer variables with values 10 and 20, and add them using a method from the Integer class. (Hint: Use Integer.sum(int, int)).**

public class Main

{

public static void main(String[] args) {

int a=10;

int b = 20;

int res = Integer.sum(a,b);

System.out.println(res);

}

}

o/p = 30;

**j. Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class. (Hint: Use Integer.min(int, int) and Integer.max(int, int)).**

public class Main

{

public static void main(String[] args) {

int a=10;

int b = 20;

int res = Integer.min(a,b);

System.out.println(res);

}

}

o/p =10;

**k. Declare an integer variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Integer class. (Hint: Use Integer.toBinaryString(int), Integer.toOctalString(int), and Integer.toHexString(int)).**

public class Main

{

public static void main(String[] args) {

int a=20;

String octal= Integer.toOctalString(a);

String hex= Integer.toHexString(a);

String binary= Integer.toBinaryString(a);

System.out.println(octal+ " "+hex+" "+binary);

}

}

o/p = 24 14 10100;

**l. Experiment with converting an int value into other primitive types or vice versa and observe the results.**

**Integer to long**

public class Main

{

public static void main(String[] args) {

int a = 42;

long b = a;

System.out.println(b);

}

}

**Integer to double**

public class Main

{

public static void main(String[] args) {

int a= 12;

double b = a;

System.out.println(b);

}

}

**Long to int**

public class Main

{

public static void main(String[] args) {

long a = 1234567890L;

int b = (int) a;

System.out.println(b)

}

}

**Double to int**

public class Main

{

public static void main(String[] args) {

double a = 3.14;

int b = (int) a;

System.out.println(b);

}

}

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

**b. Write a program to test how many bytes are used to represent a long value using the BYTES field. (Hint: Use Long.BYTES).**

public class Main

{

public static void main(String[] args) {

long a = 1234567890L;

int longSizeInBytes = Long.BYTES;

System.out.println(longSizeInBytes);

}

}

**c. Write a program to find the minimum and maximum values of long using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Long.MIN\_VALUE and Long.MAX\_VALUE).**

public class Main

{

public static void main(String[] args) {

long minValue = Long.MIN\_VALUE;

long maxValue = Long.MAX\_VALUE;

System.out.println(minValue);

System.out.println( maxValue);

}

}

o/p = -9223372036854775808

9223372036854775807

**d. Declare a method-local variable number of type long with some value and convert it to a String using the toString method. (Hint: Use Long.toString(long)).**

public class Main

{

public static void main(String[] args) {

long a = 9876543210L;

String b = Long.toString(a);

System.out.println(a);

System.out.println(b);

}

}

o/p = 9876543210

9876543210

**e. Declare a method-local variable strNumber of type String with some value and convert it to a long value using the parseLong method. (Hint: Use Long.parseLong(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "123";

long myLong = Long.parseLong(strNumber);

System.out.println(strNumber);

System.out.println(myLong);

}

}

**f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a long value. (Hint: parseLong method will throw a NumberFormatException).**

public class Main

{

public static void main(String[] args) {

String strNumber = "A123c";

long myLong = Long.parseLong(strNumber);

System.out.println(strNumber);

System.out.println(myLong);

}

}

Exception in thread "main" java.lang.NumberFormatException: For input string: "A123c"

at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.base/java.lang.Long.parseLong(Long.java:692)

at java.base/java.lang.Long.parseLong(Long.java:817)

at Main.main(Main.java:13)

**g. Declare a method-local variable number of type long with some value and convert it to the corresponding wrapper class using Long.valueOf(). (Hint: Use Long.valueOf(long)).**

public class Main

{

public static void main(String[] args) {

long a= 1234567890L;

Long b= Long.valueOf(a);

System.out.println(a);

System.out.println(b);

}

}

1234567890

1234567890

**h. Declare a method-local variable strNumber of type String with some long value and convert it to the corresponding wrapper class using Long.valueOf(). (Hint: Use Long.valueOf(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "123";

Long a = Long.valueOf(strNumber);

System.out.println(a);

}

}

**i. Declare two long variables with values 1123 and 9845, and add them using a method from the Long class. (Hint: Use Long.sum(long, long)).**

public class Main

{

public static void main(String[] args) {

long a = 1123L;

long b = 9845L;

long c = Long.sum(a, b);

System.out.println(c);

}

}

o/p = 10968;

**j. Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class. (Hint: Use Long.min(long, long) and Long.max(long, long)).**

public class Main

{

public static void main(String[] args) {

long a = 1122L;

long b = 5566L;

long minValue = Long.min(a, b);

long maxValue = Long.max(a, b);

System.out.println(minValue +" "+maxValue);

}}

o/p = 1122 5566;

**k. Declare a long variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Long class. (Hint: Use Long.toBinaryString(long), Long.toOctalString(long), and Long.toHexString(long)).**

public class Main

{

public static void main(String[] args) {

long myLong = 7L;

String binaryString = Long.toBinaryString(myLong);

String octalString = Long.toOctalString(myLong);

String hexString = Long.toHexString(myLong);

System.out.println(binaryString);

System.out.println(octalString);

System.out.println(hexString);

}

}

111

7

7

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

**b. Write a program to test how many bytes are used to represent a float value using the BYTES field. (Hint: Use Float.BYTES).**

public class Main

{

public static void main(String[] args) {

float myFloat = 3.14f;

int floatSizeInBytes = Float.BYTES;

System.out.println(floatSizeInBytes);

}

}

o/p =4;

**c. Write a program to find the minimum and maximum values of float using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Float.MIN\_VALUE and Float.MAX\_VALUE).**

public class Main

{

public static void main(String[] args) {

float minValue = Float.MIN\_VALUE;

float maxValue = Float.MAX\_VALUE;

System.out.println(minValue);

System.out.println(maxValue);

}

}

o/p = 1.4E-45

3.4028235E38

**d. Declare a method-local variable number of type float with some value and convert it to a String using the toString method. (Hint: Use Float.toString(float)).**

public class Main

{

public static void main(String[] args) {

float a = 123.45f;

String b = Float.toString(a);

System.out.println(b);

}

}

o/p = 123;

**e. Declare a method-local variable strNumber of type String with some value and convert it to a float value using the parseFloat method. (Hint: Use Float.parseFloat(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "22";

float a = Float.parseFloat(strNumber);

System.out.println(a);

}

}

**f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a float value. (Hint: parseFloat method will throw a NumberFormatException).**

public class Main

{

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

float a = Float.parseFloat(strNumber);

System.out.println(a);

}

}

o/p = number format exception

**g. Declare a method-local variable number of type float with some value and convert it to the corresponding wrapper class using Float.valueOf(). (Hint: Use Float.valueOf(float)).**

public class Main

{

public static void main(String[] args) {

float a= 123.45f;

Float b = Float.valueOf(a);

System.out.println(b);

}

}

o/p =123.45

**h. Declare a method-local variable strNumber of type String with some float value and convert it to the corresponding wrapper class using Float.valueOf(). (Hint: Use Float.valueOf(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "987.65";

Float b = Float.valueOf(strNumber);

System.out.println(b);

}

}

o/p = 987.65

**i. Declare two float variables with values 112.3 and 984.5, and add them using a method from the Float class. (Hint: Use Float.sum(float, float)).**

public class Main

{

public static void main(String[] args) {

float a = 112.3f;

float b = 984.5f;

float sum = Float.sum(a, b);

System.out.println(sum);

}

}

o/p = 1069.8

**j. Declare two float variables with values 112.2 and 556.6, and find the minimum and maximum values using the Float class. (Hint: Use Float.min(float, float) and Float.max(float, float)).**

public class Main

{

public static void main(String[] args) {

float a = 112.2f;

float b = 556.6f;

float minValue = Float.min(a, b);

float maxValue = Float.max(a, b);

System.out.println(minValue);

System.out.println(maxValue);

}

}

o/p = 112.2

556.6

**k. Declare a float variable with the value -25.0f. Find the square root of this value. (Hint: Use Math.sqrt() method).**

public class Main

{

public static void main(String[] args) {

float a = 25.0f;

double squareRoot = Math.sqrt(a);

System.out.println(squareRoot);

}

}

o/p = 5.0;

**l. Declare two float variables with the same value, 0.0f, and divide them. (Hint: Observe the result and any special floating-point behavior).**

public class Main

{

public static void main(String[] args) {

float a = 0.0f;

float b = 0.0f;

float result = a / b;

System.out.println(result);

}

}

o/p = NaN;

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

**b. Write a program to test how many bytes are used to represent a double value using the BYTES field. (Hint: Use Double.BYTES).**

public class Main

{

public static void main(String[] args) {

int a = Double.BYTES;

System.out.println(a);

}

}

o/p = 8;

**c. Write a program to find the minimum and maximum values of double using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Double.MIN\_VALUE and Double.MAX\_VALUE).**

public class Main

{

public static void main(String[] args) {

double minValue = Double.MIN\_VALUE;

double maxValue = Double.MAX\_VALUE;

System.out.println(minValue);

System.out.println(maxValue);

}

}

4.9E-324

1.7976931348623157E308

**d. Declare a method-local variable number of type double with some value and convert it to a String using the toString method. (Hint: Use Double.toString(double)).**

public class Main

{

public static void main(String[] args) {

double a = 123.45;

String b = Double.toString(a);

System.out.println(b);

}

}

o/p = 123.45;

**e. Declare a method-local variable strNumber of type String with some value and convert it to a double value using the parseDouble method. (Hint: Use Double.parseDouble(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "456.78";

double a = Double.parseDouble(strNumber);

System.out.println(a);

}

}

o/p = 456.78;

**f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a double value. (Hint: parseDouble method will throw a NumberFormatException).**

public class Main

{

public static void main(String[] args) {

String strNumber = "Ab12Cd3";

double a = Double.parseDouble(strNumber);

System.out.println(a);

}

}

o/p = number format exception

**g. Declare a method-local variable number of type double with some value and convert it to the corresponding wrapper class using Double.valueOf(). (Hint: Use Double.valueOf(double)).**

public class Main

{

public static void main(String[] args) {

double a = 789.0;

Double b = Double.valueOf(a);

System.out.println(b);

}

}

o/p = 789.0;

**h. Declare a method-local variable strNumber of type String with some double value and convert it to the corresponding wrapper class using Double.valueOf(). (Hint: Use Double.valueOf(String)).**

public class Main

{

public static void main(String[] args) {

String strNumber = "123.456";

Double a = Double.valueOf(strNumber);

System.out.println(a);

}

}

o/p = 123.456;

**i. Declare two double variables with values 112.3 and 984.5, and add them using a method from the Double class. (Hint: Use Double.sum(double, double)).**

public class Main

{

public static void main(String[] args) {

double num1 = 112.3;

double num2 = 984.5;

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

System.out.println(sum);

}

}

o/p = 1096.8;

**j. Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class. (Hint: Use Double.min(double, double) and Double.max(double, double)).**

public class Main

{

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(min);

System.out.println(max);

}

}

o/p= 112.2

556.6

**k. Declare a double variable with the value -25.0. Find the square root of this value. (Hint: Use Math.sqrt() method).**

public class Main

{

public static void main(String[] args) {

double a = -25.0;

double squareRoot = Math.sqrt(a);

System.out.println(squareRoot);

}

}

o/p =NaN;

**l. Declare two double variables with the same value, 0.0, and divide them. (Hint: Observe the result and any special floating-point behavior).**

public class Main

{

public static void main(String[] args) {

double num1 = 0.0;

double num2 = 0.0;

double result = num1 / num2;

System.out.println(result);

}

}

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

public class Main {

public static void main(String[] args) {

int a = 11;

String str1 = Integer.toString(a);

String str2 = String.valueOf(a);

System.out.println(str1);

System.out.println(str2);

long b = 123456789L;

String str3 = Long.toString(b);

String str4 = String.valueOf(b);

System.out.println(str3);

System.out.println(str4);

double c = 3.14159;

String str5 = Double.toString(c);

String str6 = String.valueOf(c);

System.out.println(str5);

System.out.println(str6);

boolean d = true;

String str7 = Boolean.toString(d);

String str8 = String.valueOf(d);

System.out.println(str7);

System.out.println(str8);

char ch = 'A';

String str9 = Character.toString(ch);

String str10 = String.valueOf(ch);

System.out.println(str9);

System.out.println(str10);

float e = 2.71828f;

String str11 = Float.toString(e);

String str12 = String.valueOf(e);

System.out.println(str11);

System.out.println(str12);

}

}

o/p =

11

11

123456789

123456789

3.14159

3.14159

true

true

A

A

2.71828

2.71828

**Q10 Calculator**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("num1: ");

int num1 = sc.nextInt();

System.out.print("num2: ");

int num2 = sc.nextInt();

System.out.print("Enter operator : +, -, \*, /");

char operator = sc.next().charAt(0);

int result = 0;

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

if (num2 != 0) {

result = num1 / num2;

} else {

System.out.println("Error: Division by zero is not allowed.");

return;

}

break;

default:

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

return;

}

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

}

}

**Q9 Defaults values of each datatypes**

public class PrimitiveDefaultsExample {

private int a;

private long b;

private double c;

private boolean val;

private char ch;

private byte d;

private short s;

private float f;

public static void main(String[] args) {

PrimitiveDefaultsExample example = new PrimitiveDefaultsExample();

System.out.println("int: " + example.a);

System.out.println("long: " + example.b);

System.out.println("double: " + example.c);

System.out.println("boolean: " + example.val);

System.out.println("char: " + example.ch);

System.out.println("byte: " + example.d);

System.out.println("short: " + example.s);

System.out.println("float: " + example.f);

}

}

o/p = int: 0

long: 0

double: 0.0

boolean: false

char:

byte: 0

short: 0

float: 0.0