**Note**: Consider the following before starting the assignment:

* A **static field** declared inside a class is called a **class-level variable**. To access this variable, use the class name and the dot operator (e.g., Integer.MAX\_VALUE).
* A **static method** defined inside a class is called a **class-level method**. To access this method, use the class name and the dot operator (e.g., Integer.parseInt()).
* When accessing static members within the same class, you do not need to use the class name.

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

**a.** Explore the [Java API documentation for java.lang.Boolean](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html) and observe its modifiers and super types.

**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) ).

**Solution—**

**package** com.example;

**import** java.lang.Boolean;

**public** **class** Program {

**public** **static** **void** main(String[] args) {

Boolean status= **true**;

String s= Boolean.*toString*(status);

System.***out***.println(s);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Boolean;

**public** **class** Program {

**public** **static** **void** main(String[] args) {

String strStatus= "true";

Boolean b= Boolean.*parseBoolean*(strStatus);

System.***out***.println(b);

}

}

**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").

**Solution-**

**package** com.example;

**import** java.lang.Boolean;

**public** **class** Program {

**public** **static** **void** main(String[] args) {

String strStatus= "0";

Boolean b= Boolean.*valueOf*( strStatus == "1" );

System.***out***.println(b);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Boolean;

**public** **class** Program {

**public** **static** **void** main(String[] args) {

Boolean status= **true**;

BOOLEAN b= Boolean.*valueOf*(status );

System.***out***.println(b);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Boolean;

**public** **class** Program {

**public** **static** **void** main(String[] args) {

String strStatus= "true";

Boolean b= Boolean.*valueOf*(strStatus);

System.***out***.println(b);

}

}

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

**Solution-**

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

**a.** Explore the [Java API documentation for java.lang.Byte](https://docs.oracle.com/javase/8/docs/api/java/lang/Byte.html) and observe its modifiers and super types.

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

**Solution-**

**package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** a= Byte.***BYTES***;

System.***out***.println(a);

}

}

**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).

**Solution-**

**package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** a= Byte.***MAX\_VALUE***;

System.***out***.println(a);

**int** b=Byte.***MIN\_VALUE***;

System.***out***.println(b);

}

}

**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)).

Solution-

**package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**byte** number= 65;

String s=Byte.*toString*(number);

System.***out***.println(s);

}

}

**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)).

**Soluiton-**

**package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="127";

Byte b= Byte.*parseByte*(strNumber);

System.***out***.println(b);

}

}

**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).

**Solution-** **package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="ABC127";

Byte b= (**byte**) strNumber.hashCode();

System.***out***.println(b);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

Byte number=127;

BYTE b= Byte.*valueOf*(number);

System.***out***.println(b);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Byte;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="-87";

Byte b= Byte.*valueOf*(strNumber);

System.***out***.println(b);

}

}

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

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

**a.** Explore the [Java API documentation for java.lang.Short](https://docs.oracle.com/javase/8/docs/api/java/lang/Short.html) and observe its modifiers and super types.

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

**Solution-**

**package** com.example;

**import** java.lang.Short;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**short** sh=56;

Byte b= Short.***BYTES***;

System.***out***.println(b);

}

}

**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).

**Solution-**

**package** com.example;

**import** java.lang.Short;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println("Max Value:"+Short.***MAX\_VALUE*** );

System.***out***.println("Max Value:"+Short.***MIN\_VALUE*** );

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Short;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**short** number =123;

String s= Short.*toString*(number);

System.***out***.println(s);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Short;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber ="124";

**short** s= Short.*parseShort*(strNumber);

System.***out***.println(s);

}

}

**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).

**Solution-**

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Short;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**short** number =12;

**Short** s= Short.*valueOf*(number);

System.***out***.println(s);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Short;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String number =12;

**short** s= Short.*valueOf*(number);

System.***out***.println(s);

}

}

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

**Solution-**

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

**a.** Explore the [Java API documentation for java.lang.Integer](https://docs.oracle.com/javase/8/docs/api/java/lang/Integer.html) and observe its modifiers and super types.

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

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Integer.***BYTES***);

}

}

**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).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Integer.***MAX\_VALUE***);

System.***out***.println(Integer.***MIN\_VALUE***);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** number= 12;

String s=Integer.*toString*(12);

System.***out***.println(s);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="12";

**int** n=Integer.*parseInt*(strNumber);

System.***out***.println(n);

}

}

**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).

**Solution-**

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** n=10;

**Integer** no= Integer.*valueOf*(n);

System.***out***.println(no);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber s= "14";

**int** n= Integer.valueOf(s);

System.out.println(n);

}

}

**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)).

Solution-

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** n1=10;

**int** n2=20;

**int** sum= Integer.*sum*(n1,n2);

System.***out***.println(sum);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** n1=10;

**int** n2=20;

**int** min= Integer.*min*(n1,n2);

System.***out***.println(min);

**int** max= Integer.*max*(n1,n2);

System.***out***.println(max);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Integer;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**int** n1=7;

String bin= Integer.*toBinaryString*(n1);

System.***out***.println(bin);

String oct= Integer.*toOctalString*(n1);

System.***out***.println(oct);

String hex =Integer.*toHexString*(n1);

System.***out***.println(hex);

}

}

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

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

**a.** Explore the [Java API documentation for java.lang.Long](https://docs.oracle.com/javase/8/docs/api/java/lang/Long.html) and observe its modifiers and super types.

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

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Long.***BYTES***);

}

}

**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).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Long.***MIN\_VALUE***);

System.***out***.println(Long.***MAX\_VALUE***);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**long** number=12;

String s=Long.*toString*(number);

System.***out***.println(s);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="12";

**long** l=Long.*parseLong*(number);

System.***out***.println(l);

}

}

**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).

**Solution-**

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**long** number=14;

Long l=Long.*valueOf*(number);

System.***out***.println(l);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber= "123";

Long l=Long.*valueOf*(strNumber);

System.***out***.println(l);

}

}

1. 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)).

Solution-

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**long** l1=1123, l2=9845 ;

**long** sum=Long.*sum*(l1,l2);

System.***out***.println(sum);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**long** l1=1123, l2=5566 ;

**long** min=Long.*min*(l1,l2);

System.***out***.println(min);

**long** max=Long.*max*(l1,l2);

System.***out***.println(max);

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Long;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**long** l= 7 ;

String bin=Long.*toBinaryString*(l);

System.***out***.println(bin);

String hex=Long.*toHexString*(l);

System.***out***.println(hex);

String oct=Long.*toOctalString*(l);

System.***out***.println(oct);

}

}

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

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

**a.** Explore the [Java API documentation for java.lang.Float](https://docs.oracle.com/javase/8/docs/api/java/lang/Float.html) and observe its modifiers and super types.

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

**Solution-**

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Float.***BYTES***);

}

}

**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).

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Float.***MIN\_VALUE***);

System.***out***.println(Float.***MAX\_VALUE***);

}

}

**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)).

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**float** number = 12.5f;

System.***out***.println(Float.*toString*(number));

}

}

**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)).

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber = "12.5";

System.***out***.println(Float.*parseFloat*(strNumber));

}

}

**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).

**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)).

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**float** number = 12.59f;

System.***out***.println(Float.*valueOf*(number));

}

}

**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)).

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**float** number = 12.59f;

System.***out***.println(Float.*valueOf*(number));

}

}

1. 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)).

package com.example;

import java.lang.Float;

public class Program{

public static void main(String[] args) {

float n1 = 112.3f, n2=984.5f;

System.out.println(Float.sum(n1,n2));

}

}

**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)).

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**float** n1 = 112.2f, n2=556.6f;

System.***out***.println(Float.*max*(n1,n2));

}

}

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

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**float** n1 = -25.0f;

System.***out***.println(Math.*sqrt*(n1));

}

}

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

**package** com.example;

**import** java.lang.Float;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**float** n1 = 0.0f, n2=0.0f;

**float** n3= (n1/n2);

System.***out***.println(n3);

}

}

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

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

**a.** Explore the [Java API documentation for java.lang.Double](https://docs.oracle.com/javase/8/docs/api/java/lang/Double.html) and observe its modifiers and super types.

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

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**MIN\_VALU**

System.***out***.println(Double.***BYTES***);

}

}

**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).

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

System.***out***.println(Double.***MAX\_VALUE***);

System.***out***.println(Double.***MIN\_VALUE***);

}

}

**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)).

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**double** number=123.45f;

System.***out***.println(Double.*toString*(number));

}

}

**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)).

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="123.45";

System.***out***.println(Double.*parseDouble*(strNumber));

}

}

**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).

**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)).

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

**double** number=123.45f;

System.***out***.println(Double.*valueOf*(number));

}

}

**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)).

**Solution-**

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

String strNumber="23.45";

System.***out***.println(Double.*valueOf*(strNumber));

}

}

**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)).

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

Double n1=112.2d,n2=556.6d;

System.***out***.println(Double.*sum*(n1,n2));

}

}

**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)).

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

Double n1=112.2d,n2=556.6d;

System.***out***.println(Double.*max*(n1,n2));

System.***out***.println(Double.*min*(n1,n2));

}

}

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

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

Double n1=-25.0d;

System.***out***.println(Math.*sqrt*(n1));

}

}

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

**package** com.example;

**import** java.lang.Double;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

Double n1=0.0d , n2=0.0d;

System.***out***.println(n1/n2);

}

}

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

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

Initialize a variable of each primitive type with a user-defined value and convert it into String:

* + First, use the toString method of the corresponding wrapper class. (e.g., Integer.toString()).
  + Then, use the valueOf method of the String class. (e.g., String.valueOf()).

Solution--

package com.example;

import java.lang.Number;

public class Program{

public static void main(String[] args) {

Double d=0.0d;

int n2=12;

float f=23.4f;

long l=123456;

short s=12;

boolean b= true;

byte by= 10;

System.out.println(Integer.toString(n2));

System.out.println(Double.toString(d));

System.out.println(Float.toString(f));

System.out.println(Short.toString(s));

System.out.println(Long.toString(l));

System.out.println(Boolean.toString(b));

System.out.println(String.valueOf(n2));

System.out.println(String.valueOf(b));

System.out.println(String.valueOf(s));

System.out.println(String.valueOf(f));

System.out.println(String.valueOf(d));

System.out.println(String.valueOf(l));

}

}

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

Declare variables of each primitive type as fields of a class and check their default values. (Note: Default values depend on whether the variables are instance variables or static variables).

**package** com.example;

**public** **class** Program{

Double d;

**int** n2;

**float** f;

**long** l;

**short** s;

**boolean** b;

**byte** by;

**char** c;

**static** Double *d1*;

**static** **int** *n23*;

**static** **float** *f1*;

**static** **long** *l1*;

**static** **short** *s1*;

**static** **boolean** *b1*;

**static** **byte** *by1*;

**public** **static** **void** main(String[] args) {

Program p=**new** Program();

System.***out***.println(p.n2);

System.***out***.println(p.d);

System.***out***.println(p.s);

System.***out***.println(p.b);

System.***out***.println(p.by);

System.***out***.println(p.l);

System.***out***.println(p.f);

System.***out***.println(p.*d1*);

System.***out***.println(p.*n23*);

System.***out***.println(p.*s1*);

System.***out***.println(p.*b1*);

System.***out***.println(p.*by1*);

System.***out***.println(p.*l1*);

System.***out***.println(p.*f1*);

}

}

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

Write a program that accepts two integers and an arithmetic operator (+, -, \*, /) from the command line. Perform the specified arithmetic operation based on the operator provided. (Hint: Use switch-case for operations).

**package** com.example;

**import** java.util.Scanner;

**public** **class** Program{

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.print("Enter number1: ");

**int** num1=sc.nextInt();

System.***out***.print("Enter number2: ");

**int** num2=sc.nextInt();

System.***out***.print("Enter the operator: ");

**char** c= sc.next().charAt(0);

**switch**(c)

{

**case** '+':

System.***out***.print("Result: "+(num1 + num2));

**break**;

**case** '-':

System.***out***.print("Result: "+(num1 -num2));

**break**;

**case** '\*':

System.***out***.print("Result: "+(num1 \*num2));

**break**;

**case** '/':

**if** (num2 == 0)

System.***out***.print("Error: Division by zero");

**else**

System.***out***.print("Result: "+(num1/num2));

**break**;

**default**:

System.***out***.print("invalid option");

}

}

}