UNIT 9: THE JAVA.LANG.MATH CLASS

UNIT STRUCTURE

9.1 Learning Objectives

9.2 Introduction

9.3 About java.lang.Math class

9.3.1 Methods of the java.lang.Math Class

9.3.2 Program Examples

9.4 Let Us Sum Up

9.5 Further Reading

9.6 Answer to Check Your Progress

9.7 Model Questions

9.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- describe the necessity of java.lang.Math class
- use various methods under java.lang.Math class
- use various logarithmic methods
- use various angular and trigonometric methods

9.2 INTRODUCTION

Java language provides various packages and built-in classes for use as per the requirements. In the last unit of the first block, the concept of thread and creating a thread was elaborately discussed. In this unit, we will explain about java.lang.Math class and various methods under this class. These methods are used for various mathematical operations. The next unit explores the concept of exception handling.

9.3 ABOUT JAVA.LANG.MATH CLASS

The java.lang package defines classes that are fundamental to the Java language. For this reason, all classes are in java.lang packages are imported automatically, so there is no reason to write an import statement

for them. The java.lang package also contains the Math class, which is used to perform basic mathematical operations such as the elementary exponential, logarithm, square root, and trigonometric functions. The Math class defines two mathematical constants PI and E. Their meaning is as follows:

- **static double E** - This is the double value that is closer than any other to e, the base of the natural logarithms.
- **static double PI** - This is the double value that is closer than any other to pi, the ratio of the circumference of a circle to its diameter.

Declaration of these two constants is:

```
public final static double Math.PI
public final static double Math.E
```

Because all methods of the Math class are defined as static, it doesn't need to create an instance to use them. It's not possible to create an instance of the Math class because the constructor is private. Further, we can't extend the Math class either, because it's marked final.

9.3.1 Methods of the java.lang.Math Class

Java.lang.Math class provides several methods to work on mathematical operations like min(), max(), avg(), sin(), cos(), tan(), round(), ceil(), floor(), abs(),sqrt(), etc. This class inherits methods from java.lang.Object class.

The methods of the Math class are static and are accessed through the class name like any static method. For these method calls, the general form is:

```
result = Math.aStaticMathMethod();
```

The following table enlists the various Math methods and their functions.

Table 9.1: Various methods under Math class

Method Name	Description
Math.abs()	Returns the Absolute value of the given value.

Math.max()	Returns the Largest of two values.
Math.min()	Finds the Smallest of two values.
Math.round()	used to round of the decimal numbers to the nearest value.
Math.sqrt()	Returns the square root of a number.
Math.cbrt()	used to return the cube root of a number.
Math.pow()	Returns the value of first argument raised to the power to second argument.
Math.signum()	used to find the sign of a given value.
Math.ceil()	Finds the smallest integer value that is greater than or equal to the argument or mathematical integer.
Math.copySign()	used to find the Absolute value of first argument along with sign specified in second argument.
Math.nextAfter()	Returns the floating-point number adjacent to the first argument in the direction of the second argument.
Math.nextUp(dd)	It returns the floating-point value adjacent to d in the direction of positive infinity.
Math.nextDown(d)	It returns the floating-point value adjacent to d in the direction of negative infinity.
Math.floor()	used to find the largest integer value which is less than or equal to the argument and is equal to the mathematical integer of a double value.
Math.floorDiv()	finds the largest integer value that is less than or equal to the algebraic quotient.
Math.random()	returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
Math.rint()	returns the double value that is closest to the given argument and equal to mathematical integer.
Math.hypot()	Computessqrt(x^2 +y^2) without intermediate overflow or underflow.
Math.getExponent()	returns the unbiased exponent used in the representation of a value.
Math.IEEEremainder()	used to calculate the remainder operation on two arguments as prescribed by the IEEE 754 standard and returns value.

Math.addExact()	returns the sum of its arguments, throwing an exception if the result overflows an int or long.
Math.subtractExact()	returns the difference of the arguments, throwing an exception if the result overflows an int.
Math.multiplyExact()	used to return the product of the arguments, throwing an exception if the result overflows an int or long.
Math.incrementExact()	returns the argument incremented by one, throwing an exception if the result overflows an int.
Math.decrementExact()	returns the argument decremented by one, throwing an exception if the result overflows an int or long.
Math.negateExact()	returns the negation of the argument, throwing an exception if the result overflows an int or long.
Math.toIntExact()	returns the value of the long argument, throwing an exception if the value overflows an int.

Moreover, Math class includes some logarithmic, angular and trigonometric functions as listed below.

Logarithmic Math Methods

Method Name	Description
Math.log()	It returns the natural logarithm of a double value.
Math.log10()	It is used to return the base 10 logarithm of a double value.
Math.log1p()	It returns the natural logarithm of the sum of the argument and 1.
Math.exp()	It returns E raised to the power of a double value, where E is Euler's number and it is approximately equal to 2.71828.
Math.expm1()	It is used to calculate the power of E and subtract one from it.

Trigonometric Math Methods

Method Name	Description
Math.sin()	It is used to return the trigonometric Sine value of a Given double value.

Math.cos()	It is used to return the trigonometric Cosine value of a Given double value.
Math.tan()	It is used to return the trigonometric Tangent value of a Given double value.
Math.asin()	It is used to return the trigonometric Arc Sine value of a Given double value
Math.acos()	It is used to return the trigonometric Arc Cosine value of a Given double value.
Math.atan()	It is used to return the trigonometric Arc Tangent value of a Given double value.

Angular Math Methods

Method Name	Description
Math.toDegrees	It is used to convert the specified Radians angle to an equivalent angle measured in Degrees.
Math.toRadians	It is used to convert the specified Degrees angle to an equivalent angle measured in Radians.

Some of the listed methods are briefly described in the following.

1. **abs()**

   The abs() method returns the absolute value of any type of argument i.e. int, long, float, or double.

- The result is positive zero if the argument is positive zero or negative zero.
- The result is positive infinity if the argument is infinite.
- The result is NaN (Not a Number) if the passed argument is NaN.

Syntax:

```
public static datatype abs(datatype arg)
```

For example, `x = Math.abs(99); // output is 99`

```
x = Math.abs(-99) // output is 99 l
```

2. **ceil()**

   The ceil() method returns the smallest possible value that is greater than or equal to the argument and equal to the nearest

integer value. In other words, the argument is rounded up to the nearest integer equivalent.

- The result is the same if the returned value is already a mathematical integer.
- The result is the same if the passed argument is NaN or infinite or zero.
- The result is negative zero if the passed argument is less than zero but greater than -1.0

Syntax:

```
public static double ceil(double arg);
```

All the following calls to Math.ceil() return the double value 9.0:

```
Math.ceil(9.0) // result is 9.0
Math.ceil(8.8) // rises to 9.0
Math.ceil(8.02) // still rises to 9.0
Math.ceil(-9.0) //return -9.0
```

3. floor()

The floor() method returns the closest integer value that is less than or equal to the argument. This method is the reverse of the ceil() method.

Syntax:

```
public static double floor(double a);
```

All the following calls to Math.floor() return the double value 9.0:

```
Math.floor(9.0) // result is 9.0
Math.floor(9.4) // drops to 9.0
Math.floor(9.8) // still drops to 9.0
```

4. max()

The max() method takes two numeric arguments and returns the greater of the two. This method is overloaded to handle int, long, float, or double arguments. If the input parameters are the same, max() returns a value equal to the two arguments.

Syntax:

```
public static datatypemax(arg1, arg2);
```

For example: `s=Math.max(23,60)//returns 60`

`S= Math.max(56.54, 25.0) // returns 56.54`

5. min()

The min() method is the opposite of the max() method. It takes two numeric arguments and returns the lesser of the two. This method is overloaded to handle int, long, float, or double arguments. If the input parameters are the same, min() returns a value equal to the two arguments.

Syntax:

```
public static datatypemin(datatypearg1, datatype
arg2);
```

For example, `x = Math.min(0.5, 0.0); // output is 0.0`

6. random()

The random() method returns a random double that is greater than or equal to 0.0 and less than 1.0. The random() method does not take any parameters.

Syntax:

```
public static double random( );
```

7. round()

The round() method returns the integer closest to the argument. The algorithm is to add 0.5 to the argument and truncate to the nearest integer equivalent. This method is overloaded to handle a float or a double argument.

Syntax:

```
public static intround(float a);
public static long round(double a);
```

8. sin()

The sin() method returns the sine of an angle. The argument is a double representing an angle in radians. Degrees can be converted to radians by using Math.toRadians().

Syntax:

```
public static double sin(double a);
```

For example, `Math.sin(Math.toRadians(90.0))// returns 1.0`

9. cos()

The cos() method returns the cosine of an angle. The argument is a double representing an angle in radians.

Syntax:

```
public static double cos(double a);
```

For example, `Math.cos(Math.toRadians(0.0)) // returns 1.0.`

10. tan()

The tan() method returns the tangent of an angle. The argument is a double representing an angle in radians.

Syntax:

```
public static double tan(double a);
```

For example, `Math.tan(Math.toRadians(45.0)) // returns 1.0.`

11. sqrt()

The sqrt() method returns the square root of a number in double. This square root function returns a complex number (comprised of real and imaginary parts) when the operand is negative.

The Java Math.sqrt() method returns NaN instead of an object representing a complex number.

Syntax:

```
public static double sqrt(double a);
```

For example, `Math.sqrt(9.0) // returns 3.0`

12. toRadians()

This method converts argument (degree) to radians.

Syntax:

```
public static double toRadians(double deg);
```

13. log()

This method returns the logarithmic value of the passed argument.

Syntax:

```
public static double log(double arg);
```

14. atan()

This method returns the arc tangent of the argument value. The returned angle is in the range -pi/2 through pi/2. arc tan is inverse tan of the argument passed.

Syntax:

```
public static double atan(double a);
```

15. asin()

This method returns the arc sine value of the argument passed. The returned angle is in the range -pi/2 to pi/2. arc sine is inverse sine of the argument passed.

Syntax:

```
public static double asin(double arg);
```

16. cbrt()

This method returns the cube root of the passed argument.

Syntax:

```
public static double cbrt(double arg);
```

9.3.2 Program Examples

The application of the above methods is shown in the following examples.

Example 9.1: Program showing the use of java math classes like max, sqrt, pow, log, and exp

```
public class JavaMathExample1
{
public static void main(String[] args)
{
double x = 60;
double y = 3;
// return the maximum of two numbers
System.out.println("Maximum number of x and y is:
```

```
" +Math.max(x, y));
// return the square root of y
System.out.println("Square root of y is: " +
Math.sqrt(y));
//returns x to power of 3
System.out.println("Power of x and y is: " +
Math.pow(x, y));
// return the logarithm of given value
System.out.println("Logarithm of x is: " +
Math.log(x));
System.out.println("Logarithm of y is: " +
Math.log(y));
// return the logarithm of given value when base
is 10
System.out.println("log10 of x is: " +
Math.log10(x));
System.out.println("log10 of y is: " +
Math.log10(y));
// return the log of x + 1
System.out.println("log1p of x is: "
+Math.log1p(x));
// return a power of 2
System.out.println("exp of a is: " +Math.exp(x));
// return (a power of 2)-1
System.out.println("expm1 of a is: "
+Math.expm1(x));
}
}
```

Example 9.2: Program showing the use of java math class like sin, cos, tanh

```
public class JavaMathExample2
{
```

```
public static void main(String[] args)
{
 double a = 30;
 // converting values to radian
  double b = Math.toRadians(a);
// return the trigonometric sine of a
   System.out.println("Sine value of a is: "
+Math.sin(a));
 // return the trigonometric cosine value of a
   System.out.println("Cosine value of a is: "
+Math.cos(a));
 // return the trigonometric tangent value of a
   System.out.println("Tangent value of a is: "
+Math.tan(a));
  // return the trigonometric arc sine of a
   System.out.println("Sine value of a is: "
+Math.asin(a));
 // return the trigonometric arc cosine value of a
  System.out.println("Cosine value of a is: "
+Math.acos(a));
 // return the trigonometric arc tangent value of
a
  System.out.println("Tangent value of a is: "
+Math.atan(a));
// return the hyperbolic sine of a
      System.out.println("Sine value of a is: "
+Math.sinh(a));
// return the hyperbolic cosine value of a
  System.out.println("Cosine value of a is: "
+Math.cosh(a));
// return the hyperbolic tangent value of a
System.out.println("Tangent value of a is: "
+Math.tanh(a));
```

```
}
}
```

Example 9.3: Program showing use of java math class like abs(), acos(), toRadians

```
// Java program explaining lang.Math class methods
// abs(), acos(), toRadians()
import java.lang.*;
public class NewClass
{
    public static void main(String[] args)
    {
        // Declaring the variables
        int Vali = -1;
        float Valf = .5f;

        // Printing the values
        System.out.println("Initial value of int
: "+Vali);
        System.out.println("Initial value of int
: "+Valf);
 // Use of .abs() method to get the absoluteValue
        int Absi = Math.abs(Vali);
        float Absf = Math.abs(Valf);
        System.out.println("Absolute value of int
: "+Absi);
       System.out.println("Absolute value of int
: "+Absf);
        System.out.println("");
        // Use of acos() method
        // Value greater than 1, so passing NaN
        double Acosi = Math.acos(60);
       System.out.println("acos value of Acosi :
```

```
"+Acosi);
        double x = Math.PI;
       // Use of toRadian() method
        x = Math.toRadians(x);
        double Acosj = Math.acos(x);
       System.out.println("acos value of Acosj :
"+Acosj);
    }
}
```

Example 9.4: Program showing use of java math class like asin() and cbrt()

```
// Java program explaining lang.Math class methods
asin(), cbrt()

import java.lang.*;
public class NewClass
{
public static void main(String[] args)
{
int a = 1, b = 8;
int radd = a+b;
       // Use of asin() method
        // Value greater than 1, so passing NaN
        double Asini = Math.asin(radd);
System.out.println("asin value of Asini : " +
Asini);
        double x = Math.PI;
       // Use of toRadian() method
        x = Math.toRadians(x);
        double Asinj = Math.asin(x);
System.out.println("asin value of Asinj : " +
Asinj);
```

```
System.out.println("");
        // Use of cbrt() method
        double cbrtval = Math.cbrt(216);
System.out.println("cube root : " + cbrtval);
    }
}
```



CHECK YOUR PROGRESS

Q 1. Choose the correct options from the followings:

i) Which of these classes is a superclass of all other classes?

a) Math b) Process c) System d) Object

ii) What is the value of double constant 'E' defined in Math class?

a) approximately 3 b) approximately 3.14

c) approximately 2.72 d) approximately 0

iii) Which of these classes contains only floating point functions?

a) Math b) Process c) System d) Object

iv) What will be the output of the following Java program?

```
class Main
{
public static void main(String args[])
{
double a=3.1;
double b=6.7;
double c=Math.max(a,b);
System.out.print (c);
    }
}
```

a) 4.5 b) 6.7 c) 3.1 d) None of these

v) What is the value of "d" in the following Java code snippet?

double m=Math.round(2.5 + Math.random());

a) 2 b) 3 c) 4 d) 2.5

vi) Which of the following would compile without error?

a) int a = Math.abs(-5); b) int a = Math.abs(5.0);

c) int a = Math.abs(5.5F); d) int a = Math.abs(5L);

vii) Which of the following are valid calls to Math.max?

1.Math.max(2,7) 2. Math.max(4.5,7)

3. Math.max(2,3,5) 4. Math.max(-4.5f,-9.0f)

a) 1,2,3 b) 2,3,4 c) 1,2,4 d) 3,4

viii) To declare a constant MAX_LENGTH as a member of the class, we have to write as:

a) final static MAX_LENGTH = 99.98;

b) final static float MAX_LENGTH = 99.98;

c) static double MAX_LENGTH = 99.98;

d) final static double MAX_LENGTH = 99.98;



9.4 LET US SUM UP

- The java.lang package defines classes that are fundamental to the Java language.
- All classes in the java.lang package are imported automatically, so there is no reason to write an import statement for them.
- The java.lang package also contains the Math class, which is used to perform basic mathematical operations.
- Java.lang.Math class provides several methods to work on mathematical operations.
- Various methods under Math class include min(), max(), avg(), sin(), cos(), tan(), round(), ceil(), floor(), abs(), etc.
- The methods of the Math class are static and are accessed like any static method-through the class name. For these method calls, the general form is: res = Math.aStaticMathMethod();



9.5 FURTHER READING

1) Schildt, H. (2014). Java: The Complete Reference, Oracle Press.

2) Rao, R. N. (2006). Core and Advanced Java Black Book, Dreamtech Press.

9.6 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1. i) d ii) c iii) a iv) b v) b vi) b vii) c viii) d

9.7 MODEL QUESTIONS

Q 1. What are the constants available in java.lang.Math class? Explain them with example.

Q 2. Write the usability of java.lang.Math class.

Q 3. Explain various methods of java.lang.Math class which are inherited from Object class.

Q 4. Write a Java program to demonstrate the use of three trigonometric functions.

*** ***** ***