|  |
| --- |
| **Set 5: Methods of the Math Class** |

|  |
| --- |
| **Skill 5.1: Use the java Math class to perform mathematical operations**  **Skill 5.2: Apply the *random()* method to create a random integer in a specified range**  **Skill 5.3: Familiarize yourself with additional methods in the java Math class** |

|  |
| --- |
| **Skill 5.1: Use the java Math class to perform mathematical operations** |

**Skill 5.1 Concepts**

The math class is a powerful class of methods for performing mathematical computations. An example of how the math class can be applied is illustrated below. The example below computes the square root of 17. The result is assigned to the double variable p because square roots typically do not result in integer values.

|  |  |
| --- | --- |
| **Code** | **Output** |
| double p = Math.sqrt(17);  System.out.println(p); | 4.123105625617661 |

In the above example,

* *double p* is the variable to which the result of the Math operation is assigned
* *Math.* is the notation we use to access the library of Math functions in java
* *square(17)* is the operation we want to perform on the number 17. In this case, it is the square root.

A close-up of a math class

AI-generated content may be incorrect.

Java provides an extensive library of Math operations. Below is a description of some of them. Notice that each method has a corresponding *signature*. The *signature* of a method can be interpreted as follows,

A diagram of a diagram

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Below are examples of each of the methods described above,

|  |  |
| --- | --- |
| **Code** | **Output** |
| double d = -379.22;  System.out.println(Math.abs(d)); | 379.22 |
| double d = 42.01;  double e = 3.728;  System.out.println(Math.pow(d,e)); | 1126831.027 |
| double d = 2034.56;  System.out.println(Math.sqrt(d)); | 45.106097148833435 |
| double d = 1.4;  System.out.println(Math.ceil(d)); | 2.0 |
| double d = -1.6;  System.out.println(Math.ceil(d)); | -1 |
| double d = 1.4;  System.out.println(Math.floor(d)); | 1.0 |
| double d = -1.6;  System.out.println(Math.floor(d)); | -2.0 |
| double d = 7.89;  System.out.println(Math.log(d)); | 2.065596134857783  NOTE: log is base e |
| double x = 2038.5;  double y = -8999.0;  System.out.println(Math.min(x, y)); | -8999.0 |
| double x = 2038.5;  double y = -8999.0;  System.out.println(Math.max(x, y)); | 2038.5 |
| double a = 148.2;  long aResult = Math.round(a);  System.out.println(Math.round(aResult));  float b = 148.7f;  int bResult = Math.round(b);  System.out.println(Math.round(bResult)); | 148  149  NOTE: The Math.round method in Java returns different datatypes depending on the argument:   * If you pass a float, it returns an int. * If you pass a double, it returns a long. |
| System.out.println(Math.PI); | 3.141592653589793 |
| System.out.println(Math.random()); | prints a random double between 0 and 1 where 0 is inclusive, but 1 is not inclusive |

**[Skill 5.1 Exercise 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set35/Set35TicketOutTheDoorAPCompSciPrinciples.pdf)**

|  |
| --- |
| **Skill 5.2: Apply the *random()* method to create a random integer in a specified range** |

**Skill 5.2 Concepts**

Many applications you will create will require a random number. For example, what if you needed to write a program to generate a number that represented a face from a 6-sided die, or a card from a 52 card deck?

The *random()* method in the Math class generates a random double type number between 0 and 1, where 0 is inclusive, but 1 is not. The below code is illustrative,

|  |  |
| --- | --- |
| **Code** | **Output** |
| System.out.println(Math.random()); | prints a random double between 0 and 1 where 0 is inclusive, but 1 is not inclusive |

To create a number in a different range, say 0 up to 10, simply multiply the result of Math.random() by the desired range. An example is shown below,

|  |  |
| --- | --- |
| **Code** | **Output** |
| System.out.println(Math.random()\*10); | prints a random double from 0 up to 10 |

Recall, however that the random() method returns a double. The below code illustrates how to generate a random int from 0 up to 10,

|  |  |
| --- | --- |
| **Code** | **Output** |
| int randomNumber = (int)(Math.random()\*10);  System.out.println(randomNumber); | prints a random int from 0 up to 10 |

The above examples, illustrate how to scale the random() method to a specified range. The example below illustrates how to shift the range of the random number.

|  |  |
| --- | --- |
| **Code** | **Output** |
| int randomNumber = (int)(Math.random()\*10)+100;  System.out.println(randomNumber); | prints a random int from 100 up to 110 |

**[Skill 5.2 Exercise 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set35/Set35TicketOutTheDoorAPCompSciPrinciples.pdf)**

|  |
| --- |
| **Skill 5.3: Familiarize yourself with additional methods in the java Math class** |

**Skill 5.3 Concepts**

In addition to the methods above, the java Math class has many more useful methods.

A screenshot of a computer

AI-generated content may be incorrect.

**[Skill 5.3 Exercises 1](https://hpluska.github.io/APCompSciPrinciples/ticketOutTheDoor/set35/Set35TicketOutTheDoorAPCompSciPrinciples.pdf)**