Java Math Functions – Questions & Answers

A. Multiple Choice Questions

1. What is the output of `Math.pow(Math.sqrt(4),2)`?
a) 2.0
b) 4.0 ✓
c) 8.0
d) None of these

---

2. How many arguments can be passed into `Math.max()`?
a) 1
b) 2 ✓
c) 3
d) 0

---

3. How many arguments can be passed into `Math.random()`?
a) 0 ✓
b) 1
c) 2
d) 3

---

4. What is the output of `Math.max(-3.5,2.3)+Math.min(2.4,1.4)`?
a) 2.3
b) 3.7 ✓
c) 12
d) 4

---

5. Which of the following returns the cube root of a number?
a) Math.pow()
b) Math.random()
c) Math.cbrt() ✓
d) Math.abs()

---

6. Which of the following is the return type of `Math.round()`?
a) boolean
b) double

c) int ✓
d) None

---

7. What will be the output of `Math.rint(3.4)`?
a) 4.0
b) 3.0 ✓
c) 3.4
d) None of these

---

8. What is the output of
```
System.out.println(Math.max(Math.min(4.5,7.4),5.3));
```
a) 4.5
b) 5.3 ✓
c) 7.4
d) None of these

---

9. Name the mathematical function which is used to find cosine of an angle in radians.
a) Math.cos() ✓
b) Math.pow()
c) Math.cosine()
d) None of these

---

10. Which Java expression represents (a+b)3|a−b|\sqrt{\frac{(a+b)^3}{|a-b|}}|a−b|(a+b)3?
a) (a+b,3)/Math.abs(a-b)
b) Math.sqrt((a+b),3)/Math.abs(a-b))
c) Math.sqrt((Math.pow(a+b),3)/(a-b))
d) Math.sqrt((Math.pow(a+b,3))/Math.abs(a-b)) ✓

B. Fill in the Blanks

1. The output of `Math.max(4.5,5)` is **5.0**
2. The output of `Math.pow(4,2)` is **16.0**
3. Return type of `Math.sqrt()` is **double**
4. The result of `Math.pow(2,3)+Math.sqrt(4)` is **10.0**
5. Java expression of √(a²+b²) → **Math.sqrt(Math.pow(a,2)+Math.pow(b,2))**
6. Java expression of (2ab√(ab)c) → **2*abc*Math.sqrt(a*b)**
7. The output of `Math.cbrt(27)` is **3.0**
8. The output of `Math.round(2.4)` is **2**
9. The output of `Math.min(-3.4,-2.3)+Math.max(-3.4,-2.1)` is **-5.5**
10. The return type of `Math.rint()` is **double**

C. Short Answer Questions

1. **What is a library method?**
   → A predefined method provided by Java library (e.g., Math.sqrt()).
2. **What are the two types of methods in Java?**
   → Library methods & User-defined methods.
3. **Examples of Math methods:**
   - `Math.max(5,9)` → 9
   - `Math.pow(2,3)` → 8.0
4. **How many arguments are required for `Math.sqrt()`?**
   → 1 argument.
5. *Write expression for A = P(1+r/n)^(rt).***
   → `A = P * Math.pow(1 + r/n, r*t)`
6. **Difference between `Math.ceil()` and `Math.floor()`.**
   - ceil(x): rounds UP to nearest integer.
   - floor(x): rounds DOWN to nearest integer.
7. **Difference between library method and user-defined method.**
   - Library method: built-in (e.g., Math.pow()).
   - User-defined: created by programmer.
8. **Java expression for $\sqrt{(2a^2+u^2)}$.**
   → `Math.sqrt(2*a*a + u*u)`
9. **Output of `Math.sqrt(9)`.**
   → 3.0
10. **Output of `System.out.println(Math.floor(-0.88));`.**
    → -1.0

✓ D. More Unsolved Programs

Here are **Java codes** for each program 👇

---

1. Maturity amount

```
import java.util.*;
class Maturity {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Principal: ");
    double P = sc.nextDouble();
    System.out.print("Enter Rate of Interest: ");
    double r = sc.nextDouble();
    System.out.print("Enter Time (years): ");
    double t = sc.nextDouble();

    double amount = P * Math.pow((1 + r/100), t);
    System.out.println("Maturity Amount = " + amount);
  }
}
```

2. Surface area & volume of a cone

```
import java.util.*;
class Cone {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    double pi = Math.PI;
    System.out.print("Enter radius: ");
    double r = sc.nextDouble();
    System.out.print("Enter slant height: ");
    double l = sc.nextDouble();
    System.out.print("Enter height: ");
    double h = sc.nextDouble();

    double surfaceArea = (pi * r * l) + (pi * r * r);
    double volume = (pi * r * r * h) / 3;

    System.out.println("Surface Area = " + surfaceArea);
    System.out.println("Volume = " + volume);
  }
}
```

---

3. Pendulum time period

```
import java.util.*;
class Pendulum {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    double pi = Math.PI;
    System.out.print("Enter length: ");
    double l = sc.nextDouble();
    System.out.print("Enter gravity: ");
    double g = sc.nextDouble();

    double T = 2 * pi * Math.sqrt(l / g);
    System.out.println("Time Period = " + T);
  }
}
```

---

4. Formula $r = \sqrt[3]{a} + b^2 - \sqrt[3]{c}$

```
import java.util.*;
class Formula {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a: ");
    double a = sc.nextDouble();
    System.out.print("Enter b: ");
    double b = sc.nextDouble();
    System.out.print("Enter c: ");
    double c = sc.nextDouble();

    double r = Math.cbrt(a) + Math.pow(b,2) - Math.cbrt(c);
    System.out.println("Result = " + r);
  }
}
```

5. Area & circumference of circle (given diameter)

```
import java.util.*;
class Circle {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    double pi = Math.PI;
    System.out.print("Enter diameter: ");
    double d = sc.nextDouble();
    double r = d / 2;

    double area = pi * r * r;
    double circumference = 2 * pi * r;

    System.out.println("Area = " + area);
    System.out.println("Circumference = " + circumference);
  }
}
```

---

6. Population growth (Pe^rt)

```
import java.util.*;
class Population {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    double e = 2.71828;
    System.out.print("Enter starting population P0: ");
    double P0 = sc.nextDouble();
    System.out.print("Enter rate r: ");
    double r = sc.nextDouble();
    System.out.print("Enter time t (years): ");
    double t = sc.nextDouble();

    double population = P0 * Math.pow(e, r*t);
    System.out.println("Population after " + t + " years = " + population);
  }
}
```

A. Tick (✓) the correct answer

1. `Math.pow(Math.sqrt(4),2)`

- `Math.sqrt(4)` → 2.0
- `Math.pow(2.0,2)` → 4.0

✓ b. 4.0

---

2. How many arguments can be passed into `Math.max()`?

- Overloaded method: 2 arguments always.

✓ 2

3. How many arguments can be passed into `Math.random()`?

- No arguments.
  ✓ a. 0

---

4. `Math.max(-3.5,2.3)+Math.min(2.4,1.4)`

- `Math.max(-3.5,2.3)` = 2.3
- `Math.min(2.4,1.4)` = 1.4
- Sum = 3.7
  ✓ 3.7

---

5. Cube root of a number → `Math.cbrt()`
✓ c. Math.cbrt()

---

6. Return type of `Math.round()` → `long` (for double input) / `int` (for float input).
Since general case → **int**
✓ c. int

---

7. `Math.rint(3.4)` → rounds to nearest integer as double → 3.0
✓ b. 3.0

---

8. `System.out.println(Math.max(Math.min(4.5,7.4),5.3));`

- `Math.min(4.5,7.4)` = 4.5
- `Math.max(4.5,5.3)` = 5.3
  ✓ 5.3

---

9. Cosine of an angle in radians → `Math.cos()`
✓ a. Math.cos()

---

10. Expression (a+b)3/|a−b|(a+b)^3 / |a-b|(a+b)3/|a−b| inside square root →
✓ d. Math.sqrt((Math.pow(a + b, 3)/Math.abs(a - b)))

✓ B. Fill in the blanks

1. `Math.max(4.5,5)` → **5.0**
2. `Math.pow(4,2)` → **16.0**
3. Return type of `Math.sqrt()` → **double**
4. `Math.pow(2,3)+Math.sqrt(4)` = 8 + 2 = **10.0**
5. Java expression of $\sqrt{(a^2+b^2)}$ → **Math.sqrt(Math.pow(a,2)+Math.pow(b,2))**
6. Expression of $2abc\sqrt{ab}$ → **2*abc*Math.sqrt(a*b)**
7. `Math.cbrt(27)` → **3.0**
8. `Math.round(2.4)` → **2**
9. `Math.min(-3.4,-2.3)+Math.max(-3.4,-2.1)` = -3.4 + -2.1 = **-5.5**
10. Return type of `Math.rint()` → **double**

✓ C. Short Answer Questions

1. **Library method** → Predefined method provided by Java libraries (e.g., `Math.sqrt()`).
2. **Two types of methods** → Library methods & User-defined methods.
3. Examples:
   - `Math.max(5,9)` → 9
   - `Math.pow(2,3)` → 8.0
4. Arguments for `Math.sqrt()` → **1 argument**.
5. Expression: `A = P * Math.pow(1 + r/n, r*t)`
6. `Math.ceil(x)` → rounds UP to nearest integer.
   `Math.floor(x)` → rounds DOWN to nearest integer.
7. **Library method**: built-in, reusable (e.g., `Math.sqrt()`).
   User-defined: written by programmer for specific need.
8. Expression $\sqrt{(2a^2+u^2)}$ → `Math.sqrt(2*a*a + u*u)`
9. `Math.sqrt(9)` → 3.0
10. `Math.floor(-0.88)` → -1.0