

**Moon is looping on a
28 day cycle**

Understanding Java basics

Let's print the phases of the moon - [Moon.java \(Run online\)](#)

The screenshot displays the Online Java Compiler interface. The top navigation bar includes the logo, "Online Java Compiler", and links for "Execute", "Beautify", "Share", "Source Code", and "Help". The "Execute" button is highlighted with a red box.

The main editor area contains the following Java code:

```
1 public class Moon {  
2  
3     int phase;  
4  
5     int day;  
6  
7  
8     public void showMoon(int phase) {  
9         switch (phase) {  
10            case 1 -> {  
11                if (day == 0) {  
12                    System.out.println();  
13                }  
14                System.out.print("(");  
15            }  
16            case 2 -> System.out.print("|");  
17            case 3 -> System.out.print(")");  
18            case 4 -> System.out.print("");  
19        }  
20    }  
21  
22    int getPhase(int day){  
23        this.day = day % 28;  
24        if (this.day < 7) {  
25            phase = 1;  
26        } else if (this.day < 14){  
27            phase = 2;  
28        } else if (this.day < 21){  
29            phase = 3;  
30        } else {  
31            phase = 4;  
32        }  
33    }  
34 }
```

The right-hand side shows the output in the Terminal window. It displays four rows of ASCII art representing moon phases, each preceded by its corresponding symbol: "(" for Phase 1, "|" for Phase 2, ")" for Phase 3, and "" for Phase 4. A large red arrow points from the terminal output down to a legend box.

The legend box contains the following mapping:

(-	Phase 1
	-	Phase 2
)	-	Phase 3
	-	Phase 4

Execution flow of the Class Moon

```
public class Moon {
```

```
    7 usages  
    int phase;
```

```
    5 usages  
    int day;
```

}

```
    1 usage  
    public void showMoon(int phase) {  
        switch (phase) {  
            case 1 -> {  
                if (day == 0) {  
                    System.out.println();  
                }  
                System.out.print("(");  
            }  
            case 2 -> System.out.print("|");  
            case 3 -> System.out.print("|");  
            case 4 -> System.out.print(")");  
        }  
    }
```

}

```
    1 usage  
    int getPhase(int day){  
        this.day = day % 28;  
        if (this.day < 7) {  
            phase = 1;  
        } else if (this.day < 14){  
            phase = 2;  
        } else if (this.day < 21){  
            phase = 3;  
        } else {  
            phase = 4;  
        }  
        return phase;  
    }
```

}

```
    1 usage  
    void runMoon(int days){  
        for(int day = 0; day < days; day++) {  
            phase = getPhase(day);  
            showMoon(phase);  
        }  
    }
```

}

```
    public static void main(String[] args) {  
        Moon moon = new Moon();  
        int days = 100;  
        moon.runMoon(days);  
        //new line  
        System.out.println("\n");  
    }
```

}

Global Variables
(to use in all the methods)

showMoon method
(accepts phase, prints the
moon symbol of that phase)

getPhase method
(accepts day, finds which
phase that day falls in and
returns it)

runMoon method
(accepts no. of days,
calls getPhase and passes it to
showMoon for each day)

main method
(triggers the execution, calls
runMoon passing the input as
100 days)

EXECUTION STARTS HERE



FOR LOOP

```
for ( Initial  
Number int i=0; Exit  
Condition i < 28; Increment  
Decrement i++ ) {  
  
    //code inside the loop  
  
}
```

END

“If you know Java, you know the whole IT.”