

WAP To Input The Rating Of A Movie, And Print As Per The Given Criteria: Rating Category8.0-10 Blockbuster6.0-7.9

WAP To Input The Rating Of A Movie, And Print As Per The Given Criteria:
Rating Category8.0-10 Blockbuster6.0-7.9

Introduction

In the realm of movie reviews and ratings, it is essential to categorize films based on their rating scores to help viewers make informed decisions. Ratings serve as a quick reference to gauge the quality and popularity of a movie. This article discusses how to develop a simple program (WAP – Write A Program) that takes a movie's rating as input and prints out its category according to predefined criteria.

The specific criteria are:

- Ratings from 8.0 to 10 are classified as Blockbuster.
- Ratings from 6.0 to 7.9 are classified as Good.
- Ratings below 6.0 are considered Average or Below Par.

The goal is to implement this logic efficiently using conditional statements and demonstrate the approach in a programming language or pseudocode.

Understanding the Requirements

What is the Input?

- The program should accept a numerical value representing the movie's rating.
- Ratings are typically floating-point numbers (e.g., 7.5, 8.3).

What is the Output?

- The program should output the category of the movie based on the rating:
- Blockbuster for ratings between 8.0 and 10.
- Good for ratings between 6.0 and 7.9.
- Below Average for ratings below 6.0 (optional based on criteria, but included for completeness).

Assumptions

- Ratings are valid numerical inputs.
- Ratings are within the range of 0 to 10.
- The input can be decimal (floating point).

Designing the Logic

Step-by-step Approach

1. Input the rating from the user.
2. Validate the input to ensure it falls within 0 to 10.
3. Use conditional statements to determine the category:
 - If rating ≥ 8.0 and ≤ 10 : print Blockbuster.
 - Else if rating ≥ 6.0 and < 8.0 : print Good.
 - Else if rating < 6.0 : print Below Average.
4. Display the result to the user.

Pseudocode

```
```plaintext
Start
Prompt user to enter movie rating
Read rating
If rating ≥ 8.0 and rating ≤ 10
Print "Blockbuster"
Else if rating ≥ 6.0 and rating < 8.0
Print "Good"
Else if rating < 6.0
Print "Below Average"
Else
Print "Invalid rating"
End
```
```

Sample Implementation in Different Languages

Python Implementation

```
```python
Program to categorize movie ratings

Input rating from user
rating = float(input("Enter the movie rating (0-10): "))

Check if rating is within valid range
if 0 $\leq$ rating $\leq$ 10:
 if rating $\geq$ 8.0:
 print("Blockbuster")
```

```

elif rating >= 6.0:
print("Good")
else:
print("Below Average")
else:
print("Invalid rating. Please enter a number between 0 and 10.")
```

```

Java Implementation

```

```java
import java.util.Scanner;

public class MovieRatingCategory {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the movie rating (0-10): ");
double rating = scanner.nextDouble();

if (rating >= 0 && rating <= 10) {
if (rating >= 8.0) {
System.out.println("Blockbuster");
} else if (rating >= 6.0) {
System.out.println("Good");
} else {
System.out.println("Below Average");
}
} else {
System.out.println("Invalid rating. Please enter a number between 0 and
10.");
}

scanner.close();
}
}
```

```

C++ Implementation

```

```cpp
include
using namespace std;

int main() {
double rating;
cout <cin >> rating;

if (rating >= 0 && rating <= 10) {
if (rating >= 8.0) {
cout <> else if (rating >= 6.0) {
cout <> else {

```

```
cout <>
} else {
cout <>

return 0;
}
...
```

---

## Enhancements and Additional Considerations

### Handling Edge Cases

- Ratings exactly at boundary values like 8.0, 7.9, 6.0 should be properly categorized.
- Inputs that are not numbers or outside the 0-10 range should be handled gracefully.

### User Experience Improvements

- Re-prompt the user for valid input if invalid.
- Allow multiple inputs without restarting the program.
- Provide more detailed feedback or suggestions.

### Extending Functionality

- Categorize ratings into more detailed categories such as Excellent, Average, Poor.
- Include movie titles for more contextual output.
- Integrate with a database or file system to process multiple movies.

---

## Summary

Creating a program to categorize movie ratings based on specified criteria is a fundamental exercise in conditional logic. By accepting a rating input, validating it, and applying conditional statements, the program can effectively classify movies into categories such as Blockbuster, Good, or Below Average. This approach not only aids in understanding basic programming constructs but also provides a foundation for more complex movie review systems.

The key points to remember are:

- Proper input validation.
- Correct use of conditional operators.
- Clear and user-friendly output.

Implementing such programs enhances logical thinking and prepares developers for more advanced data processing tasks.

---

## Conclusion

Developing a simple rating classification program is an excellent way to practice fundamental programming concepts. Whether using Python, Java, C++, or any other language, the core logic remains the same. The ability to input data, process conditions, and display results is essential for building more extensive systems in the future. By adhering to the criteria outlined—ratings from 8.0-10 as Blockbuster and 6.0-7.9 as Good—developers can create effective and intuitive classification tools for movies and other rating-based systems.

---

End of Article

## Frequently Asked Questions

### **How can I categorize a movie's rating as Blockbuster or Not based on the given criteria?**

You can input the movie's rating and use conditional statements to check if the rating falls between 8.0-10 for Blockbuster, 6.0-7.9 for others, and then print the appropriate category.

### **What programming language is suitable for implementing the rating categorization program?**

Languages like C, C++, Java, or Python are suitable for this task since they support input handling and conditional logic needed to classify movie ratings.

### **How do I handle invalid inputs when prompting for movie ratings?**

You should validate the input to ensure it is a numeric value within the expected range (e.g., 0 to 10). If invalid, prompt the user again or display an error message.

### **Can this program be extended to include more rating categories?**

Yes, you can add more categories by defining additional rating ranges and updating the conditional logic accordingly for more detailed classifications.

## What is the sample output for a rating of 8.5 in this program?

For a rating of 8.5, the program should output: 'Blockbuster', since it falls within the 8.0-10 range.

## Additional Resources

WAP To Input The Rating Of A Movie, And Print As Per The Given Criteria:  
Rating Category  
8.0-10 Blockbuster  
6.0-7.9

In the world of cinematic entertainment, ratings serve as crucial indicators for viewers seeking to gauge the quality and appeal of movies. Whether you're a budding programmer or an enthusiast interested in understanding basic conditional logic, crafting a simple program to categorize movie ratings can be both educational and practical. This article explores the process of creating a straightforward program—using a programming language like C, Java, or Python—that prompts the user to input a movie's rating and then classifies it based on predefined categories: "Blockbuster" for ratings between 8.0 and 10, and "Hit" for ratings between 6.0 and 7.9.

Understanding the importance of such a program is essential for beginners learning conditional statements, data validation, and user interaction. Moreover, this task exemplifies how real-world applications often depend on simple decision-making logic to provide meaningful outputs, reinforcing foundational programming concepts.

---

## Fundamentals of Rating Classification

### Defining the Rating Categories

Before diving into the coding aspect, it's crucial to understand the criteria for rating classification:

- Blockbuster: Ratings ranging from 8.0 to 10 (inclusive). These movies are considered highly successful and widely acclaimed.
- Hit: Ratings ranging from 6.0 to 7.9 (inclusive). These movies are generally favorable but may not have achieved massive blockbuster status.

It's important to note that ratings outside these ranges are not explicitly categorized in this context. However, in a comprehensive program, you might want to handle such cases as well, perhaps labeling them as "Below Expectations" or "Unrated."

## Why These Categories Matter

Categorizing movies based on ratings helps viewers quickly understand the general reception of a film. For producers and marketers, these classifications can influence promotional strategies. From a programming perspective, implementing such categorization sharpens skills in handling user input, conditional logic, and output formatting.

---

## Setting Up the Programming Environment

To implement this program, choose a language that supports user input and conditional statements. Popular options include:

- C: A foundational language for learning programming concepts.
- Java: Known for its portability and robust syntax.
- Python: User-friendly syntax, especially suitable for beginners.

For demonstration purposes, Python is often preferred due to its simplicity and readability, making it easier to focus on core logic.

Prerequisites:

- Python installed on your system (version 3.x recommended)
- A text editor or IDE (such as VS Code, PyCharm, or even Notepad++)

---

## Step-by-Step Implementation Guide

### 1. Prompting User Input

The first step involves asking the user to input the movie's rating. Since ratings are decimal numbers (e.g., 8.5), ensure that the input is accepted as a floating-point value.

Sample code snippet in Python:

```
```python
rating = float(input("Enter the movie rating (e.g., 7.5): "))
```
```

## 2. Validating User Input

To ensure robustness, validate that the input falls within a logical range, typically between 0.0 and 10.0. If the input is invalid, prompt the user again or display an error message.

```
```python
if rating < 0 or rating > 10:
    print("Invalid rating. Please enter a value between 0 and 10.")
    Optionally, repeat input prompt
```
```

## 3. Applying Conditional Logic

Based on the rating, use if-elif-else statements to determine the category.

```
```python
if 8.0 <= rating <= 10:
    category = "Blockbuster"
elif 6.0 <= rating < 8.0:
    category = "Hit"
else:
    category = "Unrated or Below Expectations"
```
```

In this code, ratings outside the specified ranges are categorized separately. You may customize this as needed.

## 4. Displaying the Result

Finally, output the classification to the user:

```
```python
print(f"The movie is rated as: {category}")
```
```

Putting it all together, the complete program might look like this:

```
```python
Program to classify movie ratings

Input user rating
rating = float(input("Enter the movie rating (0.0 - 10.0): "))

Validate input
if rating < 0 or rating > 10:
```

```

print("Invalid rating. Please enter a value between 0 and 10.")
else:
    Determine category
    if 8.0 <= rating <= 10:
        category = "Blockbuster"
    elif 6.0 <= rating < 8.0:
        category = "Hit"
    else:
        category = "Below Expectations or Unrated"

    Output the classification
    print(f"The movie is rated as: {category}")
    ```

```

---

## Extending the Program for Practical Use

While the basic program accomplishes the core task, real-world applications often require more features:

- Handling multiple inputs: Allow users to input ratings for several movies in one session.
- Reading data from files: Process a list of ratings stored externally.
- Graphical user interface (GUI): Develop a simple GUI for user interaction.
- Database integration: Store ratings and categories for future reference.

However, for educational purposes, focusing on single inputs and straightforward conditional logic offers valuable learning opportunities.

---

## Common Challenges and Troubleshooting

Implementing such programs can pose certain challenges, especially for beginners:

- Incorrect data types: Attempting to compare string inputs directly with numeric ranges causes errors.

Solution: Always convert input strings to float or int as appropriate.

- Boundary conditions: Ensuring ratings exactly equal to 8.0 or 6.0 are correctly categorized.

Solution: Use ``<=`` and ``>=`` operators carefully to include boundary values.

- Input validation: Users might enter non-numeric values, causing runtime errors.

Solution: Use try-except blocks to handle exceptions gracefully.

```
```python
try:
    rating = float(input("Enter the rating: "))
except ValueError:
    print("Invalid input. Please enter a numerical value.")
```

```

## Conclusion

Creating a program to input movie ratings and classify them based on predefined criteria offers a practical exercise in fundamental programming concepts such as user input, conditional statements, and output formatting. By understanding and implementing these basics, learners build a strong foundation that can be extended to more complex applications like data analysis, user interfaces, and database management.

Such projects not only enhance coding skills but also deepen understanding of how simple decision-making logic can influence real-world decision processes. Whether for academic purposes, hobby projects, or professional development, mastering these fundamental techniques is an essential step toward becoming proficient in programming.

Remember, the key to success lies in careful planning, validating user input, and thoroughly testing various scenarios to ensure your program handles all cases gracefully. Happy coding!

## [Wap To Input The Rating Of A Movie And Print As Per The Given Criteria Rating Category8 0 10 Blockbuster6 0 7 9](#)

Wap To Input The Rating Of A Movie And Print As Per The Given Criteria Rating Category8 0 10 Blockbuster6 0 7 9

## Related Articles

- ["The Health Consequences Of Air Pollution In The Twentieth Century Were Gargantuan, Although Hard To](#)
- [5. Indicate The Escherichia Coli Strains Associated With Gastrointestinal Syndromes And The Major Characteristics](#)

- [What Are The Key Services Provided By Financial Intermediaries? What Is Disintermediation? What Is Cross-intermediation?](#)

[Back to Home](#)