OSR Character Generator

In OSR, character stats refer to the numerical values that represent a character's abilities and skills. These stats are typically determined by rolling dice and applying modifiers based on the character's race, class, and other factors.

The most common stats found in OSR games include:

1. Strength (STR): Measures a character's physical power and ability to carry heavy objects, perform feats of strength, and deal more damage in combat.
2. Dexterity (DEX): Represents a character's agility, reflexes, and hand-eye coordination. It influences actions like dodging attacks, picking locks, and using ranged weapons.
3. Constitution (CON): Reflects a character's overall health, stamina, and resistance to diseases, toxins, and fatigue. It affects hit points (a measure of how much damage a character can withstand before being defeated).
4. Intelligence (INT): Measures a character's mental acuity, memory, and reasoning abilities. It can influence skills such as magic use, knowledge checks, and problem-solving.
5. Wisdom (WIS): Represents a character's intuition, perception, and common sense. It can affect skills like perception, tracking, and resisting mind-affecting effects.
6. Charisma (CHA): Reflects a character's personal magnetism, charm, and leadership qualities. It can influence interactions with NPCs (non-player characters), reaction rolls, and certain social skills.

In traditional OSR games, these stats are typically determined through dice rolls, commonly using 3d6 (rolling three six-sided dice) for each stat. The rolled values are then modified based on the character's race, class, and other factors, such as magic items or temporary effects.

These stats form the foundation for various in-game actions, determining a character's capabilities, strengths, and weaknesses. They provide a framework for players to make decisions and resolve challenges within the game world.

Example code for generating a OSR character stats in Visual Basic .NET:

```
Dim Strength As Integer
Dim Dexterity As Integer
Dim Constitution As Integer
Dim Intelligence As Integer
Dim Wisdom As Integer
Dim Charisma As Integer

Randomize()

Strength = Int((18 - 8 + 1) * Rnd() + 8)
Dexterity = Int((18 - 8 + 1) * Rnd() + 8)
Constitution = Int((18 - 8 + 1) * Rnd() + 8)
Intelligence = Int((18 - 8 + 1) * Rnd() + 8)
Wisdom = Int((18 - 8 + 1) * Rnd() + 8)
Charisma = Int((18 - 8 + 1) * Rnd() + 8)

MsgBox("Strength: " & Strength & vbCrLf & _
        "Dexterity: " & Dexterity & vbCrLf & _
        "Constitution: " & Constitution & vbCrLf & _
        "Intelligence: " & Intelligence & vbCrLf & _
        "Wisdom: " & Wisdom & vbCrLf & _
        "Charisma: " & Charisma)
```

This code uses the `Randomize` function and the `Rnd` function to generate random numbers for the character's stats. The code then displays the generated stats using a `MsgBox`.

This is a very basic example and does not take into account any specific rules or characteristics of different OSR races or classes.

Here's an another example of code, this time in python, that generates stats for common character classes in a hypothetical OSR game:

```
import random

def roll_stat():
    return sum(sorted(random.randint(1, 6) for _ in range(3))[1:])
```

```
def generate_stats():
    stats = {}
    stats['STR'] = roll_stat()
    stats['DEX'] = roll_stat()
    stats['CON'] = roll_stat()
    stats['INT'] = roll_stat()
    stats['WIS'] = roll_stat()
    stats['CHA'] = roll_stat()
    return stats

def generate_stats_for_class(character_class):
    class_stats = {
        'fighter': {'STR': 15, 'DEX': 12, 'CON': 13, 'INT': 9,
'WIS': 8, 'CHA': 10},
        'rogue': {'STR': 12, 'DEX': 15, 'CON': 11, 'INT': 10,
'WIS': 8, 'CHA': 13},
        'wizard': {'STR': 9, 'DEX': 12, 'CON': 10, 'INT': 15,
'WIS': 8, 'CHA': 11}
    }
    base_stats = class_stats.get(character_class.lower())
    if not base_stats:
        return None

    stats = generate_stats()
    for stat, value in base_stats.items():
        stats[stat] = max(value, stats[stat])

    return stats

# Example usage
character_class = 'fighter'
stats = generate_stats_for_class(character_class)
if stats:
    print(f"Stats for a {character_class}: {stats}")
else:
    print(f"Invalid character class: {character_class}")
```

In this code, the `roll_stat()` function generates a random stat value by rolling three six-sided dice and summing the second and third highest values. The `generate_stats()` function calls `roll_stat()` for each stat and returns a dictionary of the generated stats.

The `generate_stats_for_class()` `function takes a character class as input and retrieves the base stats for that class from a predefined dictionary.

It then generates the remaining stats using `generate_stats()` and ensures that each stat is at least as high as its base value.

Finally, it returns the complete set of stats for the given class.

You can modify the `class_stats` dictionary to include more character classes and their respective base stat values.