

# Modern C++ Features: Programs, Tips, and Revision Notes

## 1. auto Keyword

Description: Automatically deduces the type of a variable based on its initializer.

Example Code:

```
auto a = 10;           // int
auto b = 3.14;         // double
auto c = "Hello";      // const char*
auto d = true;         // bool
```

*Tip: Use 'auto' to simplify code and avoid redundancy.*

## 2. Range-Based for Loop

Description: Simplifies looping over containers like arrays, vectors, etc.

Example Code:

```
vector<string> fruits = {"Apple", "Banana", "Mango"};
for (auto fruit : fruits) {
    cout << fruit << endl;
}
```

*Tip: Use when you don't need the index and just want the values.*

## 3. Lambda Functions

Description: Inline anonymous functions, useful for short tasks.

Example Code:

```
auto isEven = [](int x) {
    return x % 2 == 0;
};
```

*Tip: Use lambdas for simple callbacks, conditions, or transformations.*

## 4. constexpr

Description: Defines compile-time constant expressions.

Example Code:

```
constexpr int square(int x) {
    return x * x;
}
```

*Tip: Use for constant expressions like array sizes, static configs.*

## 5. std::tuple

Description: Groups multiple values of different types.

Example Code:

```
tuple<int, string, float> student = make_tuple(101, "Surya", 88.5f);
```

```
cout << get<0>(student); // 101
```

*Tip: Use tuple to return multiple values from a function.*

## 6. enum class

Description: Strongly typed enums for better safety and scoping.

Example Code:

```
enum class Status { Success, Failure, Pending };
```

```
Status s = Status::Success;
```

*Tip: Use enum class to avoid naming conflicts and ensure type-safety.*