**Slip 3: Sample Solutions and Explanations**

**Q1. Display First ‘n’ Numbers of Fibonacci Series**

**Approach**

* Use a loop to generate and print the first n terms of the Fibonacci sequence.
* Start with the first two terms (0 and 1), then compute each next term as the sum of the previous two.

**Code**

#include <iostream>  
using namespace std;  
  
// [Fibonacci Function Definition]  
void fibonacci(int n) {  
 int a = 0, b = 1;  
 cout << a << " " << b << " ";  
 for(int i = 2; i < n; ++i) {  
 int next = a + b;  
 cout << next << " ";  
 a = b;  
 b = next;  
 }  
 cout << endl;  
}  
  
int main() {  
 int n;  
 cout << "Enter number of terms: ";  
 cin >> n;  
 fibonacci(n);  
 return 0;  
}

**Explanation**

* The fibonacci function prints the first two terms, then uses a loop to calculate and print each subsequent term.
* Each new term is the sum of the previous two, and the variables are updated accordingly.
* The user specifies how many terms to print.

**Syntax Definitions**

* **void**: Indicates the function does not return a value.
* **for loop**: Used to repeat a block of code a specific number of times.

**Q2. Student Class: Print Details of Students with Grade 'O'**

**Approach**

* Define a Student class with attributes: name, age, and grade.
* Accept details for n students and store them in a vector.
* Display details of students whose grade is 'O'.

**Code**

#include <iostream>  
#include <vector>  
using namespace std;  
  
// [Student Class Definition]  
class Student {  
 string name;  
 int age;  
 char grade;  
public:  
 // [Accept Input]  
 void accept() {  
 cout << "Name: "; cin >> name;  
 cout << "Age: "; cin >> age;  
 cout << "Grade (O/A/B/C/D): "; cin >> grade;  
 }  
 // [Display Function]  
 void display() { cout << name << " " << age << " " << grade << endl; }  
 char getGrade() { return grade; }  
};  
  
int main() {  
 int n;  
 cout << "Number of students: ";  
 cin >> n;  
 vector<Student> students(n);  
 for(auto &s : students) s.accept();  
 cout << "Students with grade O:\n";  
 for(auto &s : students)  
 if(s.getGrade() == 'O') s.display();  
 return 0;  
}

**Explanation**

* The Student class encapsulates student data and provides methods to accept and display it.
* The program reads n students, then iterates through the list, displaying only those with grade 'O'.
* The getGrade method is used to check each student's grade.

**Syntax Definitions**

* **class**: A user-defined type that groups data and functions.
* **vector**: A dynamic array from the C++ Standard Library.
* **auto**: Automatically deduces the type of the variable from its initializer.

**Q3. Math Utility Library (Static Functions Case Study)**

**Approach**

* Create a class with static functions for mathematical operations (factorial and prime check).
* Call these functions directly using the class name, without creating an object.

**Code**

#include <iostream>  
using namespace std;  
  
// [MathLib Class Definition]  
class MathLib {  
public:  
 // [Static Factorial Function]  
 static int factorial(int n) {  
 int result = 1;  
 for(int i = 1; i <= n; ++i) result \*= i;  
 return result;  
 }  
 // [Static Prime Check Function]  
 static bool isPrime(int n) {  
 if(n < 2) return false;  
 for(int i = 2; i\*i <= n; ++i)  
 if(n % i == 0) return false;  
 return true;  
 }  
};  
  
int main() {  
 int n;  
 cout << "Enter number for factorial: "; cin >> n;  
 cout << "Factorial: " << MathLib::factorial(n) << endl;  
 cout << "Enter number for prime check: "; cin >> n;  
 cout << (MathLib::isPrime(n) ? "Prime" : "Not Prime") << endl;  
 return 0;  
}

**Explanation**

* The MathLib class provides static methods for factorial and prime checking.
* Static methods can be called using the class name without creating an object.
* The program demonstrates both functionalities with user input.

**Syntax Definitions**

* **static**: Declares a member function or variable that belongs to the class, not to any object.
* **bool**: A data type that can hold true or false.
* **?: (ternary operator)**: A shorthand for if-else to choose between two values.