CIF

Library (C++)

Library

Book

Member

Library manager

Book Name,

author,

ISBN

brrower id,

is available

↓

Name,

Number

member it

borrowed is bm.

So, tere the first thing is when we add a book or member isbn we have to create random number

for that a string isbn

→ String id

for Create Random isbn

String Gerbate Isbn ()

↓

gerate random 13 number string that

start with 978 Predix

start with

for that we use this code

this is the code

```cpp
string Library::genrateIsbn()
{

    static std::random_device rd;
    static std::mt19937 gen(rd());
    std::uniform_int_distribution<> dis(0, 9);

    std::string first12 = "978";
    for (int i = 0; i < 9; ++i)
    {
        first12 += std::to_string(dis(gen));
    }
    // now we suffel all of that to make sure about true random

    std::uniform_int_distribution<> ds(3, 11);

    for (int i = 11; i > 2; i--)
    {
        int rad = ds(gen);

        char temp = first12[i];
        first12[i] = first12[rad];
        first12[rad] = temp;
    }
    return first12;
}
```

this is the code to get randome num. from device

random-device rd → to get ramdom number from device hardware noice.

mt19937 gen(rd()) → this is the first method to get random number.

here it requires seed to genrate → giving rd which is Changing every single time for more random.

and this final line

Here dis(3 ; 11) → Here we set the range that

if we give input it gives us in range of [3, 11]

further dig (genc) → adding gen the Complete par dum Num. give us final ans.

and loop goes [0, 9).

1. **gen is *Pseudo-Random*, Not Truly Random:**
    - rd (the "device") is the part that tries to be **truly random** by using system noise.
    - gen (the "generator") is **pseudo-random**. It uses a mathematical formula. It *looks* random, but if you give it the same seed from rd, it will create the exact same sequence of numbers every time. It generates a single, very large raw number, not a "row."
2. **dis Makes a *Single* Digit (0-9), Not 10 Digits:**
    - This is the main misunderstanding. dis(0, 9) doesn't create "10 digits."
    - It takes the big raw number from gen and *converts* it into a **single digit** that is somewhere between 0 and 9 (inclusive).

Correct Flow

Here is the correct step-by-step flow:

1. **rd (True Random):** Provides a single, truly random **seed** (just once).
2. **gen (Pseudo-Random):** Uses that seed to get ready to produce a sequence of big, raw *pseudo-random* numbers.
3. **dis(gen) (Distribution):** We *call* gen (via dis) to get one of those big raw numbers, and dis **shapes** it into a **single digit** (e.g., 3).