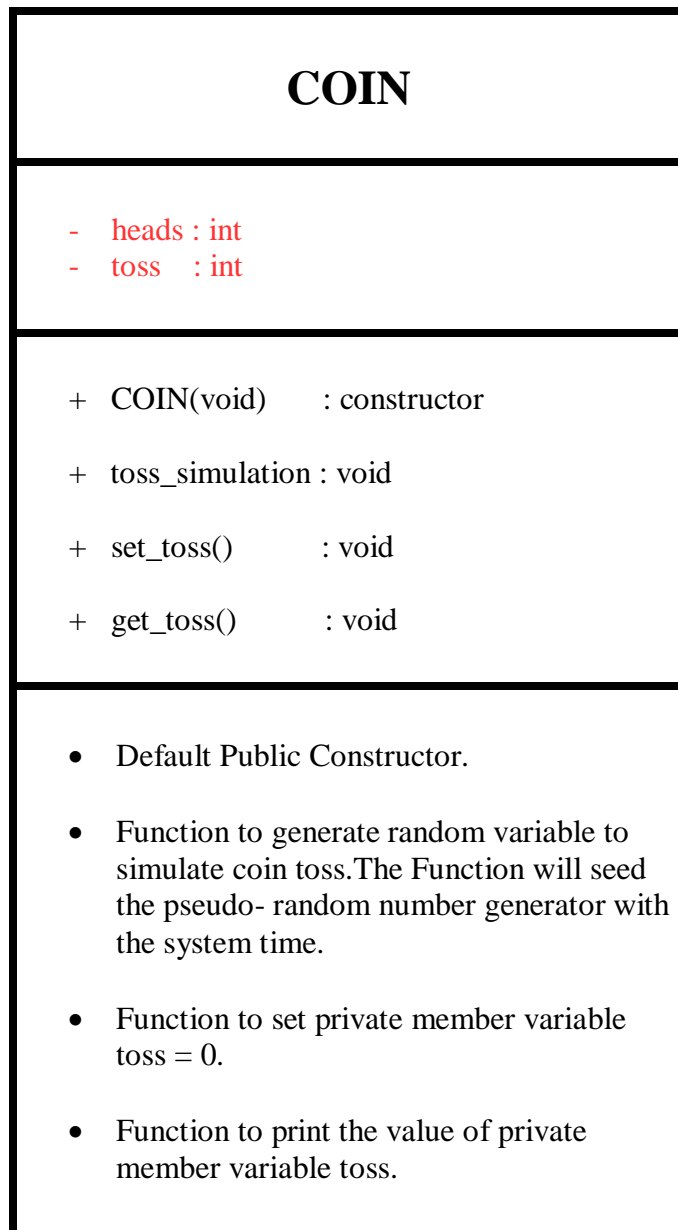


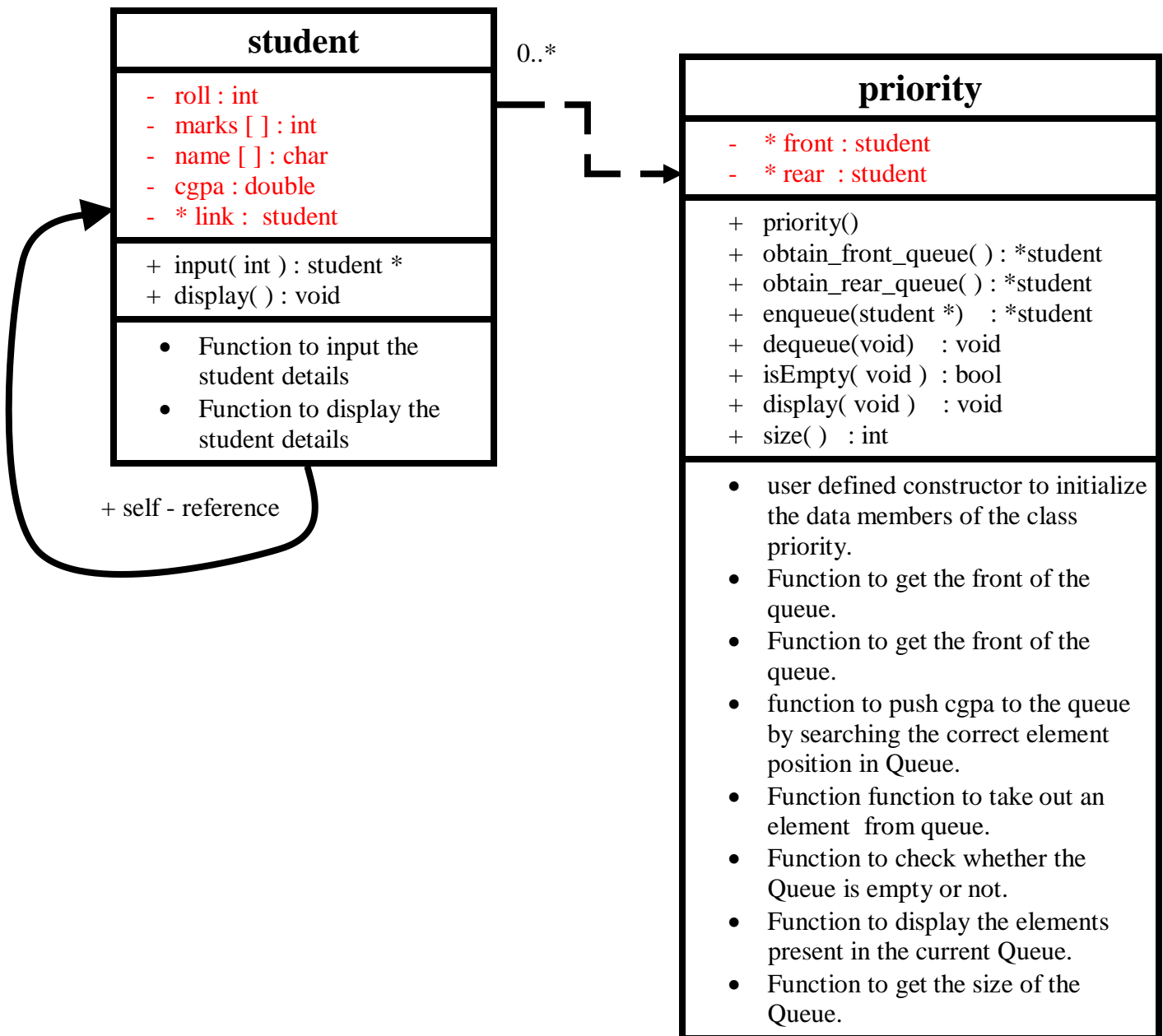
Problem (1) : Class Diagram for program that simulates flipping a coin repeatedly and continues until three consecutive heads are tossed. At that point, your program should display the total number of coin flips that were made.



Problem (2) : Class Diagram for Problem to print the pascal's triangle using Combination Function and Dynamic Programming (Using the values of elements' of previous row).

Pascal
<ul style="list-style-type: none"><li>- arr [ ] [ ] : int</li><li>- row : int</li><li>- run : int</li></ul>
<ul style="list-style-type: none"><li>- fact( int ) : int</li><li>- com( int , int ) : int</li><li>+ pascal(void) : constructor</li><li>+ generate_pascal_thru_comb( ) : void</li><li>+ generate_pascal_thru_dp() : void</li><li>+ display ( ) : void</li><li>+ check_validity( ) : int</li></ul>
<ul style="list-style-type: none"><li>• Function to return the factorial of given number n.</li><li>• Function to return the value of <math>nCk</math> given values of (n,k).</li><li>• Default Public Constructor for Initialisation.</li><li>• Function to generate pascal triangle through Combination(n,k) given the row number and Column Number.</li><li>• Function to Generate Pascal Triangle by adding the 2 elements of previous rows from current column and previous column.</li><li>• Function to display the Pascal's Triangle</li><li>• Function to check Validty of an element given row number and column number.</li></ul>

Problem (3) : Implement a Priority Queue with linked lists. Provide functionalities exactly like the template provided in C++. With Respect ot student class.



Problem(4) : Design an ARRAY class with the following features:

- Array object may be declared for a specific size and a value for initializing all the elements. If this it is to be assumed as a 0.
- An array object may be declared and initialized with another object.
- An array object may be declared and initialized with another array (not the object, standard array as in C language).

Let a and b are two objects:

- a+b will add corresponding elements.
- a=b will do the assignment.
- a[I] will return the ith element of the object.
- a\*5 or 5\*a will multiply the element with 5.

Array
<ul style="list-style-type: none"> <li>- n : int *</li> <li>- size : int</li> </ul>
<ul style="list-style-type: none"> <li>- ret_size( ) : int</li> <li>+ Array(void)</li> <li>+ operator = (int *f) : void</li> <li>+ operator = ( Array ) : void</li> <li>+ operator [ ] ( int i ) : int</li> <li>+ operator * (const int ) : Array</li> <li>+ &lt;&lt;friend&gt;&gt; operator * ( int , Array ) : Array</li> <li>+ operator + ( Array ) : Array</li> <li>+ print_array(void) : void</li> </ul>
<ul style="list-style-type: none"> <li>• Private Function to return the value of private data member size.</li> <li>• Public user defined default constructor.</li> <li>• Assignment operator overloading for Array object and integer pointer.</li> <li>• Bracket operator overloading where element with index (int) is returned.</li> <li>• Multiplication operator overloading for different syntax of multiplication with Array object.</li> <li>• Addition Operator overloading for addition of 2 array objects.</li> <li>• Function to print the array</li> </ul>

Problem (5) : Design a STRING class, which will have the initialization facility similar to array class.  
Provide support for

- ( 1 ) Assigning one object for another,
- ( 2 ) Two string can be concatenated using + operator,
- ( 3 ) Two strings can be compared using the relational operators.

## STRING

- a : char \*

```
+ STRING ( )  
+ STRING ( const STRING & )  
+ STRING ( std :: string )  
+ operator = ( const char * ) : void  
+ operator = ( char ) : void  
+ <<friend>> operator + ( STRING , STRING ) : STRING  
+ <<friend>> operator << ( std :: ostream & , STRING & ) : std :: ostream &  
+ <<friend>> operator < ( STRING , STRING ) : int  
+ <<friend>> operator > ( STRING , STRING ) : int  
+ <<friend>> operator == ( STRING , STRING ) : int  
+ scan ( void ) : void
```

- User defined Default and parameterised constructors for initialisation.
- Assignment operator overloading for character pointer initialisation.
- Addition operator overloading for 2 STRING type Object.
- Extraction Operator Overloading of ostream class for cascading of output.
- Greater than and Less than operator overloading for 2 STRING type Objects.
- Equality Operator Overloading for Comparison Checks.
- Function for scanning input.

Problem (6) : to modify the previous string program so that assigning/initializing a string by another will not copy it physically but will keep a reference count, which will be incremented. Reference value 0 means the space can be released.

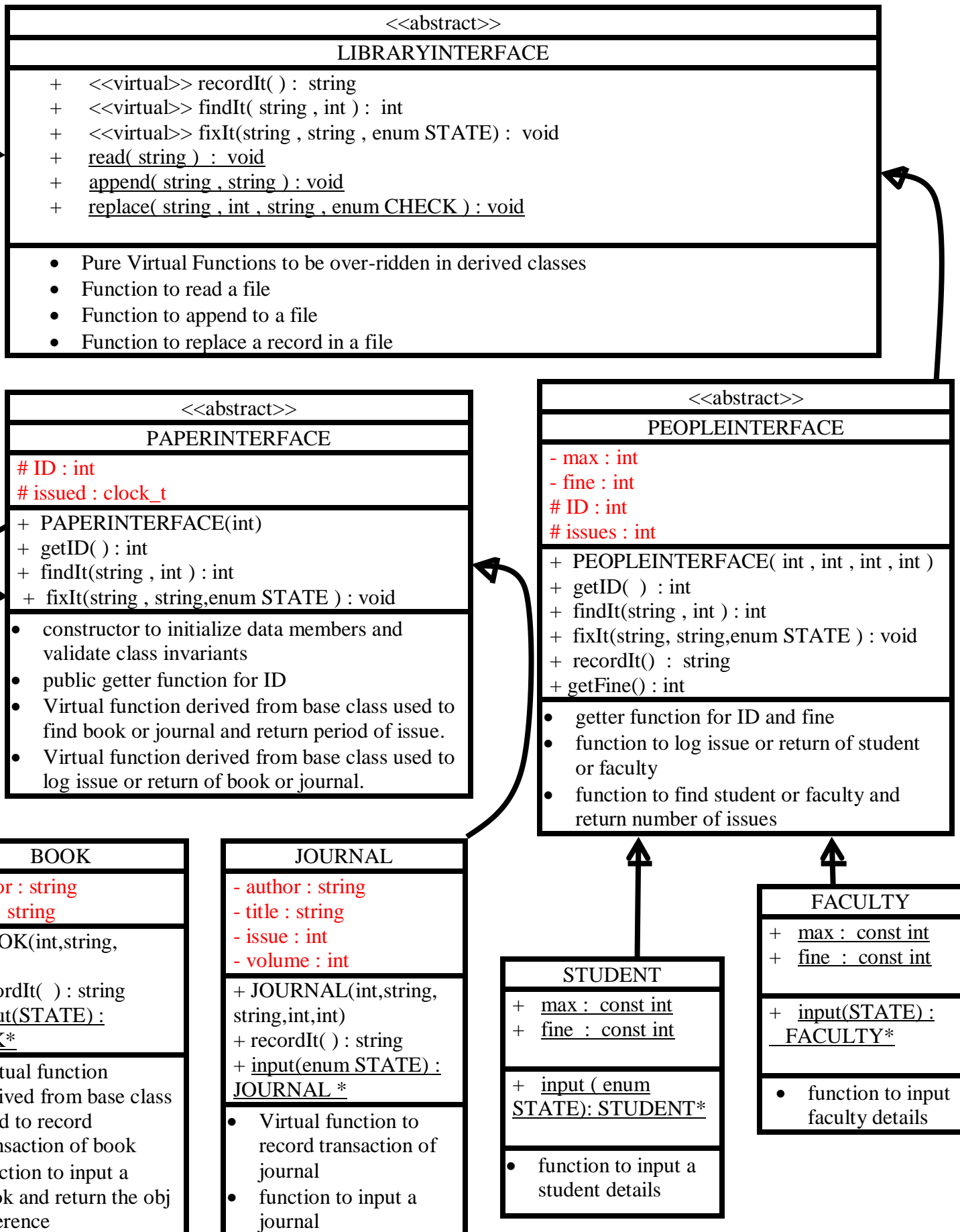
## String

- size : int
- arr : char\*
- cp : int

- + String( void )
- + String( int )
- + String(const String & )
- + String(string s)
- + display(): void
- + operator +(String) : void
- + operator =(String): String
- + operator ==(String) : bool
- + operator >(String): bool
- + operator <(String): bool

- User defined Default and parameterised constructors for initialisation.
- Function to Display the String
- Assignment operator overloading for character pointer initialisation.
- Addition operator overloading for 2 STRING type Object.
- Greater than and Less than operator overloading for 2 STRING type Objects.
- Equality Operator Overloading for Comparison Checks.

Problem( 7 ) : In a library, books, and journals are kept. Journals are issued to faculty members only. A student member can have 2 books issued at a time. For faculty members it is 10. For late return student members are charged Rs. 1 per day. Faculties are not charge. For journals additional information like issue no., date of publish, volume no., etc., are to be stored. For any transaction, members are supposed to place transactions slip. After necessary validations, transaction is carried out. Each transaction is to be noted into a register. Implement the system described above after designing the necessary classes.



Problem ( 8 ) : You are a government consultant and you are trying to figure out the best way to arrange for Internet access in your small island, Sunderban. There are N villages on your island connected by M various roads and you can walk between any two villages on the island by traversing some sequence of roads. However, you have got a limited budget and have determined that the cheapest way to arrange for internet access is to build some fiber-optic cables along existing roadways. You have a list of the costs of laying fiber-optic cable down along any particular road, and want to figure out how much money you will need to successfully complete the project “meaning that, at the end, every village will be connected along some sequence of fiber-optic cables. Luckily, you are also Sunderban's resident computer scientist, and you remember hearing about Prim's algorithm in one of your old programming classes. This algorithm is exactly the solution to your problem, but it requires a priority queue... Here's the C++ standard template library to the rescue.

Graph
<ul style="list-style-type: none"> <li>- <b>V</b> : int</li> <li>- <b>adj</b> : list&lt;pair &lt;int , int &gt; &gt; *</li> </ul>
<ul style="list-style-type: none"> <li>- &lt;&lt;friend&gt;&gt; primMST(const Graph &amp; , set &lt; int &gt; &amp;, int ) : void</li> <li>+ Graph(int)</li> <li>+ addEdge(int v,int w,int wt) : void</li> </ul>
<ul style="list-style-type: none"> <li>• friend function for implementing Prim's Algorithm</li> <li>• Constructor to initialise graph.</li> <li>• Function to add Edge to the graph</li> </ul>

heapNode
<ul style="list-style-type: none"> <li>- <b>v</b> : int</li> <li>- <b>dest</b> : int</li> <li>- <b>key</b> : int</li> </ul>
<ul style="list-style-type: none"> <li>+ &lt;&lt;friend&gt;&gt; primMST(const Graph &amp; , set &lt; int &gt; &amp;, int ) : void</li> <li>+ operator &gt; (const heapNode&amp; , const heapNode &amp; ) : bool</li> </ul>
<ul style="list-style-type: none"> <li>• Friend function for implementing prim's algorithm</li> <li>• Overloading Greater than operator to check the value of left and right key of the 2 heapNode objects passed.</li> </ul>