Problem (1): Class Diagram for TIME class which have receiving and showing facilities in any format.

#### Time

- hours : int
- minutes: int
- + input\_time\_in\_format\_12(): void
- + input\_time\_in\_format\_24() : void
- + show\_time\_in\_format\_12 () : void
- + show\_time\_in\_format\_24(): void
- + add\_time(): void
- Asks to store a time in 12 Hours(A.M./P.M.) format
- Stores Time in 24 Hours Format
- Checks for input validation and proper formatting for units mentioned while executing.
- Displays the time in user preferred format of 12hours or 24 hours.
- Performs addition with user given input in minutes.

Problem (2): Class Diagram for COMPLEX class which has accepting and display facility of a complex number after addition or subtraction of the same.

## Complex

- real: int
- comp: int
- + set(void): void
- + get(void): void
- + add(complex,complex) : void
- + subtract (complex,complex): void
- Asks to store a number in any format of complex number (a+ib/a-ib/a-ib/-ib)
- Checks for input validation and proper formatting of the complex number while executing.
- Performs addition and subtraction with user given input in complex form.
- Displays the result in the precised format of the complex number.(a+-ib/a/+-ib)

Problem (3): Class Diagram for TOLLTAX class which has accepting and display facility of toll and no of cars has been passed and amount of toll collected till now.

## Tolltax

- no\_of\_cars[]: int\*
- Toll : int
- + recieve\_toll (): void
- + display\_cars (): void
- + display\_toll () : void
- + tolltax(void) :constructor
- Asks to the passenger to pay exact amount of toll for different type of vehicles.
- Checks for input validation and proper formatting.
- Displays the total number of cars passed for each type and total amount of toll collected till now.
- User Defined Constructor for Initialising the data members to their Initial Value

Problem (4): Class Diagram for ITEM class which can add items to the list with unique id, with any number of quantities and a specified rate (which can be changed later).

#### Stock

- list[]:item
- no of items in list: static int
- + check\_exists\_return\_index(int): int
- + add\_to\_list(void) : void
- + rate\_change(void): void
- + receive(int): void
- + issue(int) : void
- + show\_price(void): void
- + show\_quantity(void): void
- Asks to add new items to the list with item code and initial quantity.
- Checks for input validation and proper formatting of the item code.
- Displays the total quantity of a particular item present in the shop and it's rate.
- Can add more item to the list and also issue item to the customer.
- Facilitates to change the price of a particular item.

#### Item

- name : char[]
- quantity : intrate : int
- item\_code : int
- + get\_name(): char\*
- + get\_item\_code():int
- + get\_rate():int
- + set\_rate (): int
- + get\_quantity(): int
- + set\_quantity(): int
- + update(): void
- Returns item\_name
- Returns item code
- Returns the price/rate of item with given Item\_code
- Updates the price/rate of item with given Item\_Code
- Returns the Quantity of item with given Item\_code
- Updates the Quantity of item with given Item\_Code
- Function To Initialise the data members of Item with the passed arguement values.

Problem (5): Class Diagram for STUDENT class which can add new student to a particular stream with proper name, auto generated id and numbers obtained in the examination to show the mark sheet of that particular one.

# Batch list []:student count: static int admit student(batch &, student &): void exists or not(batch,int): int get\_count(batch ) : int display1(batch b): void Admits students checks existence of student prints the number of student in batch Display the general details of all students in batch Date isleapyear(int) : bool valid\_date(int,int,int) : int is\_valid\_date(string) : int Function to check whether a year is leap year or not Function to check whether the int values passed as month,date, year are valid or not Function for Extracting the date from string valur passed to the

function.

# Student name[]:char roll:int date: string course: string marks[]:int sub[]: string get\_data(student &) : void display(student): void update\_roll(student &,int): void get\_roll(student) : int get\_name(student) : string get\_course(student): string change marks(student &): void Asks to add new student to the list with exact name, admission date & course. Checks for input validation and proper formatting of the above and auto generate the roll number. Asks to add the number in the particular subject associated with that particular course of that particular student.

Display the mark sheet of the student with grade as well as the

CGPA.