LAB 8

| Question | Task | TIME ALLOCATION | REMARKs |
| --- | --- | --- | --- |
| 1 | Class as a friend | 40 minutes |  |
| 2 | Class as a friend | 80 minutes |  |

**Question 1**

a) Based on the program given below, answer the questions from **(1)** to **(11).**

|  |
| --- |
| #include <iostream>  using namespace std;  class geometry  {  private:  float PI, height, radius;  public:  //-----(1)-----  //-----(2)-----    //-----(3)-----  //-----(4)-----  };  class cylinder  {  private:  float vol;  public:  //-----(5)-----    };  class cone  {  private:  float vol;  public:  //-----(6)-----  };  int main()  {  float hg, rd;  cout << "Enter height : ";  cin >> hg;  cout << "Enter radius : ";  cin >> rd;    //-----(7)-----  //-----(8)-----  //-----(9)-----  //-----(10)-----  //-----(11)-----  } |

|  |
| --- |
| **Sample Output Screen** |
| Enter height : ***3.5***  Enter radius : ***4.0***  Volume of cylinder with radius 4 and height 3.5 is : 175.952  Volume of cone with radius 4 and height 3.5 is : 58.6507  = END OF PROGRAM = |

1. Define a parameterized constructor that takes in two float parameters, *hg* and *rd*. Assigned values *3.142*, *hg* and *rd* to *PI*, *height* and *radius* respectively.
2. Define a destructor that displays on the output screen **"= END OF PROGRAM =**". **[Note: refer to sample output screen above]**
3. Declare class *cylinder* as a friend.

1. Declare class *cone* as a friend.
2. Define function *calc\_vol(…)* for class *cylinder*.

* Parameter : pointer *g* of class *geometry*
* Use pointer *g* (use dot operator [.] and indirection operator [\*]), calculate the volume of a cylinder and display the output.
* Formula : *vol* = *PI* × *radius2* × *height*

**[Note: refer to sample output screen above]**

1. Define function *calc\_vol(…)* for class *cone*.

* Parameter : reference argument *g* of class *geometry*
* Use reference argument *g*, calculate the volume of a cone and display the output.
* Formula : *vol* = × *PI* × *radius2* × *height*

**[Note: refer to sample output screen above]**

1. Declare object *gmt* of class *geometry*, passing in *hg* and *rd* as parameter values.
2. Declare object *cyc* of class *cylinder*.

1. Call object *cyc*’s *calc\_vol(…)* method, passing in address of object *gmt*.
2. Declare object *cn* of class *cone*.
3. Call object *cn*’s *calc\_vol(…)* method, passing in object *gmt*.

1. Modify the codes created in Question 1 (a) by adding a few items below in order to calculate volume of cube. **Hint** : Need changes a bit at class *geometry* and get length input.
2. Declare class *cube* as friends.
3. Define function *cal\_vol(…)*for class *cube*.
   * Parameter : reference argument *g* of class *geometry*
   * Use reference argument *g*, calculate the volume of a cube and display the output.

* Formula : *lenght3*

**[Note: refer to sample output screen below]**

1. Declare object *cb* of class *cube*.
2. Call object *cb’s calc\_vol(…)*method, passing in object *gm*t

|  |
| --- |
| **Sample Output Screen** |
| Enter height : 10  Enter radius : 30  **Enter length : 12**  Volume of cylinder with radius 30 and height 10 is : 28278  Volume of cone with radius 30 and height 10 is : 9426  **Volume of cube with length 12 is : 1728**  = END OF PROGRAM = |

**Question 2**

1. **Create class Ticket** with the following:
2. Data members (**private**):

* *no* : int
* *price* : float

1. Declare class **Student** as a *friend* of class **Ticket.**

1. Public member functions:
   * + - A **default** **constructor** to set *price* to 10.00

* void **setTickets()**

Prompt and get user input for *no* (number of tickets to purchase).

1. **Create class Student** with the following:
2. Data members (**private**):

* *id* : string
* *name* : string
* *purchase* : string
* *P* : Ticket

1. Public member functions:

* **void** **setStudent( )**

Prompt and get user input for *name* and *id.*

* **void ticket\_entry( )**

Prompt user whether to purchase tickets.

* + - If user enters ‘Y’, set *purchase* to *“Yes”,* and call function *setTickets()* using object *P*.
    - If user choose not to purchase ticket, display “--------No tickets purchase--------“
* **void display( )**

To display the student’s details (*name*, *id)* and additional details . If *purchase* equals *“Yes’*, display the details using object *P*(*no*, total price of tickets purchased (*no* multiply *price*). Otherwise, display *“You’ve not purchased any tickets”.****[Refer to sample output screen]***

1. In **main( ):**

* Declare an array of 3 object elements of class **Student.**
* Using a for-loop that loops on every array element:
* Call **setStudent( )**, **ticket\_entry( )**, and **display( )**

|  |
| --- |
| **Sample Output Screen 1 b) #1** |
| Enter ID : ***1011***  Enter Name : ***Julia***  Do you want to purchase charity tickets? [Enter Y or N]:***Y***  Please enter number of tickets to purchase: ***2***  --------------------------------  STUDENT DETAILS  --------------------------------  ID : 1011  Name : Julia  --------------------------------  ADDITIONAL DETAILS  --------------------------------  You've purchase 2 Tickets  Total amount : RM 20  Enter ID : ***1013***  Enter Name : ***Andy***  Do you want to purchase charity tickets? [Enter Y or N]:***N***  --------No tickets purchase--------  --------------------------------  STUDENT DETAILS  --------------------------------  ID : 1013  Name : Andy  --------------------------------  ADDITIONAL DETAILS  --------------------------------  You've not purchased any tickets  Enter ID : ***1055***  Enter Name : ***Roslan***  Do you want to purchase charity tickets? [Enter Y or N]:***Y***  Please enter number of tickets to purchase: ***20***  --------------------------------  STUDENT DETAILS  --------------------------------  ID : 1055  Name : Roslan  --------------------------------  ADDITIONAL DETAILS  --------------------------------  You've purchase 20 Tickets  Total amount : RM 200  -------------------------------- |