LAB 3

| Question | Task | TIME ALLOCATION | REMARKs |
| --- | --- | --- | --- |
| 1, 2, 3 | Class operation & access control | 80 minutes |  |
| 4 | More on Classes and Local / global objects | 40 minutes |  |

**Question 1**

Write a complete program to calculate the total price of a holiday package based on the selected package and number of adults and children input by the user. Use the given class:

**class TravelPackage**

**{ public:**

**char package;**

**int noAdult, noChild;**

**float adultPrice, childPrice, discount, totalPrice;**

**};**

Use the following requirements and sample input and output to write the program.

* + - * Use *do…while*statement to prompt the user for correct package selection. If the input is incorrect display *“Invalid selection”* and prompt again for package selection.
* Use *switch*statement to determine the price based on the available package.

|  |  |
| --- | --- |
| Package | Price (RM) |
| A | Adult = 1000.00  Child = 500.00 |
| B | Adult = 800.00  Child = 600.00 |
| C | Adult = 500.00  Child = 300.00 |

* Use *if* statement todeduct 20% from the total price if the price is greater than RM3000.00.

|  |
| --- |
| **Sample Output Screen** |
| Select travel package <A,B,C> : *L*  Invalid selection.  Select travel package <A,B,C> : *A*  Enter no of adult : *4*  Enter no of children : *3*  Total price : RM4400.00 |

**Question 2**

**Scenario: Finding the Volume of a Cuboid**

*A cuboid is a 3 dimensional shape.  
So to work out the volume we need to know 3 measurements.*

|  |  |  |
| --- | --- | --- |
| https://www.mathsisfun.com/geometry/images/cuboid.gif |  | Look at this shape.  There are 3 different measurements:  Length,   Width,   Height |

1. Given the class declarations in the program below, write the complete codes for required member functions based on your analysis of the program output.

Note: Your functions may be defined inside the class or outside the class.

**#include<iostream>**

**using namespace std;**

**class Cuboid**

**{ int length, width, height, volume;**

**public:**

**void setdata();**

**void findVolume();**

**void display();**

**};**

**int main()**

**{ Cuboid Q;**

**Q.setdata();**

**Q.findVolume();**

**Q.display();**

**return 0;**

**}**

|  |
| --- |
| **Sample Output Screen** |
| Enter the width, length, height of a Cuboid object :*4 8 2*  ------Display Cuboid Data-----  Width :4 cm  Length :8 cm  Height :2 cm  Volume :64 cm^3  Press any key to continue . . . |

1. Based on your solution, include additional accessor functions to return the data members for this class. Remove the function call (**Q.display()**) in your main(), and display the data members for **Q** object by calling the accessor functions.

**Question 3**

(a) Write a complete program based on the following information.

1. Create a class called **Purchase**.
2. Data member (set to private) : ***name(char)****,* ***code(char)****,* ***qty(int)****,* ***price(float)*** and ***total(float)****.*
3. Member function (set to public) : define the member functions outside of the class.

(a) **set\_data(…)**

To set all the data to the appropriate variables.

(b) **calculate()**

Calculate the total amount of payment to be made.

(c) **print()**

Display all the information on screen.

1. In main() function, do the following:
2. Create an object of class Purchase called **p**.
3. Get inputs from the user and pass the data to method **set\_data()** so that the values can be set to the appropriate variables.
4. Call function **calculate()** to calculate the total amount to be paid by a customer.
5. Call function **print()** to display all the information as shown below.

|  |
| --- |
| **Sample Output Screen** |
| ========================  WELCOME  ========================  Enter name : Jacob  Product code : C101  Enter quantity : *2*  Enter price : RM *20*  ========================  RECEIPT  ========================  Name : Jacob  Product Code : C101  Quantity : 2  Product Price : RM 20  Total Payment : RM 40 |

(b) Modify the program above, so that it will continue to execute until there is no customer or ‘N’ to purchase another item.

(Use **while loop** for this program). The suggested output is shown below.

|  |
| --- |
| **Sample Output Screen** |
| **You have another customer to purchase item? [Y/N]: Y**  **========================**  **WELCOME**  **========================**  **Enter name : David**  **Product code : C102**  **Enter quantity : 3**  **Enter price : RM 30**  **========================**  **RECEIPT**  **========================**  **Name : David**  **Product Code : C102**  **Quantity : 3**  **Product Price : RM 30**  **Total Payment : RM 90**  **You have another customer to purchase item? [Y/N]: Y**  **========================**  **WELCOME**  **========================**  **Enter name : James**  **Product code : C103**  **Enter quantity : 1**  **Enter price : RM 100**  **========================**  **RECEIPT**  **========================**  **Name : James**  **Product Code : C103**  **Quantity : 1**  **Product Price : RM 100**  **Total Payment : RM 100**  **You have another customer to purchase item? [Y/N]: N**  **Press any key to continue** |

**Question 4**

Write a C++ program that contains:

* A constant global variable **SIZE** with value **8**.
* A class **Stationery\_Inventory** with the following:
  + Private data members : **code** ***name* (*string*); *warehouses*[SIZE] (*int*)**
  + Public member functions:
    - **display\_reverse( )**
      * Display the array elements of *warehouses* in reverse order using anylooping structure.
    - **set\_data( int\* )**
      * Get user input for code *name.*
      * The function has a pointer argument.
      * In a *for* loop, use the pointer argument to initialize *warehouses* array.
    - A global object declaration, named **hold.**
* A function named **process()** :
  + - Refer to label ‘Process()’ at sample output.
    - Get user input for 10 values that should be stored in a local array.
    - Using global object **hold**, call function **set\_data(…)**, passing the array and also call **display\_reverse()** after that.
* In the main():

1. Declare an object of the class above
2. Declare an array of 8 integer elements and initialize it with the values {5, 10, 15, 22, 20, 25, 30, 35}
3. Using the object (created at (i)), make function call to **set\_data(….)** passing the array declared at (ii).
4. Using the object (created at (i)), make function call to **display\_reverse().**
5. Call **process()**

|  |
| --- |
| **Sample Output Screen** |
| **SMART Stationery Shop**  **######################################  Item Entry**  **--------------------------------------**  **Enter Stationery code: *PT123***  **-------------------------------**  **The Inventory Info**  **-------------------------------**  **Stationery code :PT123**  **Warehouse 1 :35**  **Warehouse 2 :30**  **Warehouse 3 :25**  **Warehouse 4 :20**  **Warehouse 5 :22**  **Warehouse 6 :15**  **Warehouse 7 :10**  **Warehouse 8 :5**  **--------------------------------------**  **Colourful Book Holder Stock**  **--------------------------------------**  **Enter the stocks for 8 warehouses:**  ***30***  ***45***  ***15***  ***25***  ***50***  ***19***  **Process ( )**  ***80***  ***60***  **######################################  Item Entry**  **--------------------------------------**  **Enter Stationery code: *PT321***  **-------------------------------**  **The Inventory Info**  **-------------------------------**  **Stationery code :PT321**  **Warehouse 1 :60**  **Warehouse 2 :80**  **Warehouse 3 :19**  **Process ( )**  **Warehouse 4 :50**  **Warehouse 5 :25**  **Warehouse 6 :15**  **Warehouse 7 :45**  **Warehouse 8 :30** |

**~End~**