LAB 10

| Question | Task | TIME ALLOCATION | REMARKs |
| --- | --- | --- | --- |
| 1, 2 | Dynamic Array & New Operator | 80 minutes |  |
| 3 | Dynamic Array & New Operator | 40 minutes |  |

**Question 1**

Write a program that requires users to key-in their tuition fees and from the entered values, the program will calculate total fees of the month and the average payment to be made. The dynamic array with new operator will be used to store the elements entered by user.

**[HINT: Complete the given program with refer to the sample output]**

|  |
| --- |
| #include <iostream>  using namespace std;  int main ( )  { int subject, num;  float total\_payment,average\_payment;  float \*Fees;  /\* Display and get total subjects registered \*/  ----------  ----------  Fees= new float[subject];  for (num=0; num<subject; num++)  /\* Display and get fees charge for each class registered. Then, calculate total payment \*/  ----------  ----------  ----------  ----------  /\* Display Sept payment details; whereas fees charges entered by user will be displayed accordingly \*/  ----------  ----------  ----------  /\* calculate average payment. Later, display the total payment & minimum payment to be made \*/  ----------  ----------  ----------  return 0;  } |

|  |
| --- |
| **Sample Output Screen** |
| How many subjects you have registered: 3  Class 1 fees charge: RM 60  Class 2 fees charge: RM 50  Class 3 fees charge: Rm 70  ===== SEPTEMBER 2016 PAYMENT DETAILS =====  Fees charges you have entered: 60(RM)... 50(RM)...70(RM)...  Total payment to be made: RM 180  Minimum payment (Average of total payment): RM 60 |

**Question 2**

Write a C++ program that contains:

* A constant global variable **ITEM** with value **3**.
* A class **BeanBag** with the following:
  + Private data members : **code** ***name* (*string*); *Stock* [ITEM] (*int*)**
  + Public member functions:
    - **Latest\_BeanBagStock ( )**
      * Display the ***“#Current# Ready Stocks…….”*** with display the *code name* and the array elements of *stock* in reverse order using appropriate looping structure.
    - **BeanBag\_Details (int \* )**
      * This function for print details of ***“Stocks Checking”***
      * Get user input for code *name.*
      * The function has a pointer argument.
      * In a *for* loop, use the pointer argument to initialize *with stock* array.
    - A global object declaration, named **ready.**
* A function named **StockUpdate()** :
  + - Refer to label ‘**StockUpdate** **( )**’ at sample output.
    - Get user input for 3 values that should be stored in a dynamic array with new operator.
    - Using global object **ready**, call function **BeanBag\_Details(…)**, passing in the array and also call **Latest\_BeanBagStock()** after that.
    - Delete the dynamic array created.
* In the main():

1. Declare an object of the class above
2. Declare an array of 3 integer elements and initialize it with the values {9,7,5}
3. Using the object (created at (i)), make function call to **BeanBag\_Details (….)** passing the array declared at (ii).
4. Using the object (created at (i)), make function call to **Latest\_BeanBagStock ().**
5. Call **StockUpdate ()**

|  |
| --- |
| **Sample Output Screen** |
| COSY Bean Bag Chair Sdn. Bhd.  #######################################  Stocks Checking  ---------------------------------------  Enter Bean Bag Chair code : BB123  --------------------------------------  #Current# Ready Stocks Checking...  --------------------------------------  Code Tracing >>BB123<<  Group 1 production :5 item(s) ready  Group 2 production :7 item(s) ready  Group 3 production :9 item(s) ready  ------------------------------------------  Ready Stocks [BB789] for this Month  ------------------------------------------  Update new stocks from how many group? :3  Ready stock from group 3:18  Ready stock from group 2:20  Ready stock from group 1:9  #######################################  Stocks Checking  **StockUpdate ( )**  ---------------------------------------  Enter Bean Bag Chair code :BB789  --------------------------------------  #Current# Ready Stocks Checking...  --------------------------------------  Code Tracing >>BB789<<  Group 1 production :9 item(s) ready  Group 2 production :20 item(s) ready  Group 3 production :18 item(s) ready  Press any key to continue. |

**Question 3**

Based on the requirements and sample output given below, write a full program for **class Number*.***

1. Declare for data members as private: **no (int), index (int), sample (int)** and **numbers (int).**
2. Define default constructor and display the header of this numbers sorting activity.
3. Define ***SmallestNumb (…)*** function, pass **3** parameters (***numb [ ]***, ***element*** with integer data type and ***size*** with integer data type) to this function’s header.
   1. Initialize the parameters as below:
      * element to **index**, size to **sample** and numb[..] to **no** with element value is set as the array size.
   2. In this function, use **for-loop** as a header;

**[Hint:** for (int i=element+1; i<size; i++)**]** in finding the smallest integer from the unsorted list and return the index of the smallest integer.

1. Define ***select\_Number (…)*** function, pass **2** parameters (***numbers[ ]*** and ***size***) to this function. In this function:
   1. Declare two attributes for sorting activity called: digit (int), sort (int).
   2. Use **for-loop [Hint:** for (int i=0; i<size-1; i++)**] and** call ***SmallestNumb (…)*** function with initialize this called function as a digit value. To get the smallest number, prepare the **if (…)** condition and perform the swapping.
   3. Display the sequence of elements for each sorting cycle (Refer to the sample output).
2. In ***main ( )***,
   1. Create an object for class Number.
   2. Prompt user to enter the ***numbers to be sorted***.
   3. Create a dynamic array named ***numbers*** and the size of the array is based on the number to be sorted that entered by user.
   4. Ask user to enter the numbers to be sorted one by one based on the total numbers he wants to sort (**Hint:** For user input use the dynamic array has created).
   5. Call ***select\_Number (…)*** function.
   6. Display the final sorted list.
   7. Delete the dynamic array created.

|  |
| --- |
| **Sample Output Screen** |
| NUMBERS :: Let’s PLAY  ===========================================  How many numbers would you like to be sorted : 3  The 3 numbers are:  8 3 6  The Cycle-1 sorting: 3 8 6  The Cycle-2 sorting: 3 6 8  The final list of sorted numbers is:  3 6 8  Press any key to continue… |