Question 1

Complete the code based on the instructions below:

```
#include<iostream>
using namespace std;

class ArtificialGrass
{       private: int length, width, depth;

        public:
        void grass_data(int q, int r, int s )
        {  length = q; width = r; depth = s; }

        int grass_gauge()
        { return length * width * depth;     }
};
```

Based on the program above, modify it to include the following requirements:

a) Define a **default constructor** that initializes the *length* value to 4, *width* value to 6, and *depth* value to 3.

b) Define a **destructor** that prints the "*Deleting the artificial grass details*."

c) In the main function, prepare these following codes:
   i. Create an object of class *ArtificialGrass* using the *new* operator.
   ii. Display the artificial grass's details by invoking the appropriate functions.
   iii. Delete the *ArtificialGrass*'s object.
   iv. Display the new stock available banner.
   v. Create a dynamic array of 3 objects using the *new* operator.
   vi. Using a *for* loop, call related functions:
      - Invoke the ***grass_data(..)*** function, passing the value of the ***loop counter + 2*** for each declared argument in the parameters.
      - Invoke the ***grass_gauge()*** function to print out the values.
   vii. Delete the objects you created in (v).

Sample Output Screen

```
++++++++++++++++++++++++++++++++++++++++
        The Artificial Grass Details
++++++++++++++++++++++++++++++++++++++++
The previous Batch Set left....
Available Grass Size: 72cm [2 pieces per one set]

Deleting the artificial grass details.

========================================
        The Stocks Available...
++++++++++++++++++++++++++++++++++++++++
The Grass Size SET#1
->>8cm [2 pieaves in one set]

The Grass Size SET#2
->>27cm [2 pieaves in one set]

The Grass Size SET#3
```

```
->>64cm [2 pieaves in one set]


Deleting the artificial grass details.

Deleting the artificial grass details.

Deleting the artificial grass details.
```

Question 2

a) Create a class called ***ChoreographyMarks***
   i. Data members: ***points*** and ***total*** of float type
   ii. Member functions:
      - ***default constructor***: that initializes *total* to **30**
      - void setCgPoints( )
         - set the *points* based on user input
   iii. Class *Dancer* is a friend of this class.

b) Create a class called ***InventiveMarks***
   i. Data members: ***points*** and ***total*** of float type
   ii. Member functions:
      - ***default constructor***: that initializes *total* to **20**
      - void setInvPoints( )
         - set the *points* based on user input
   iii. Class *Dancer* is a friend of this class

c) Create a class called ***Dancer***
   i. Data members: ***name*** of string type, ***age*** of int type and ***finalpoints***, ***cpoints***, ***ipoints*** of float type
   ii. Member functions:
      - void set_details( )
         - set details (*name*, and *age*) for a *Dancer*
      - void calcFinalPoints(…, …)
         - Takes an object of *ChoreographyMarks* and an object of *InventiveMarks* as arguments.
         - This function calculates the *cpoints* (in %) and *ipoints* (in %) using the raw points of the choreography *points* and inventive *points*, and their respective *total*.
           Use this formula: Percentage Points = (Points/ Total)*100;
         - Calculates the *finalpoints* (in %) based on the raw points (addition of *points* from *choreographymarks* object and *inventivemarks* object) divide by total points (addition of total from *choreographymarks* object and *inventivemarks* object)
      - getAge( )
         - returns the *age*
      - getName( )
         - returns the *name*
      - displayScoreDetails( )
         - displays *ChoregraphyMarks*, *InventiveMarks* and Final Points, (all in %), using the appropriate data members.

d) In *main( )* function
   i. prompt user to enter number of dancers to be created
   ii. create a dynamic array of objects based on the size set by user in (i), and *new* operator

iii. create an object of *ChoregraphyMarks*
iv. create an object of *InventiveMarks*
v. using a *for* loop,
- call the appropriate method to set details for each dancer
- set the points for the choreography and Inventive object each using the appropriate method.
- Display the *name* and *age* of the dancer using the appropriate method
- Call *displayScoreDetails()* using the dynamic object element to display the details of the score

vi. delete the dynamic array of objects created in (ii).

Sample Output Screen

```
Enter number of dancers
3

:::::Details for Dancer:::::
Enter name                      : Mary Ann
Enter age                       : 21
Enter Choreography Marks [max:30]: 19
Enter Inventive Marks [max:20]   : 15

========================================
Dancer #1
========================================
Name    = Mary Ann
Age     = 21

:::::Score Board:::::
Choreography    : 63.3333
Inventive       : 75
Final score     : 68
========================================

:::::Details for Dancer:::::
Enter name                      : Jane Ng
Enter age                       : 20
Enter Choreography Marks [max:30]: 27
Enter Inventive Marks [max:20]   : 17

========================================
Dancer #2
========================================
Name    = Jane Ng
Age     = 20

:::::Score Board:::::
Choreography    : 90
Inventive       : 85
Final score     : 88
========================================

:::::Details for Dancer:::::
Enter name                      : Joanna Paul
Enter age                       : 19
Enter Choreography Marks [max:30]: 18
Enter Inventive Marks [max:20]   : 14

========================================
Dancer #3
========================================
```

```
Name    = Joanna Paul
Age     = 19

:::::Score Board:::::
Choreography    : 60
Inventive       : 70
Final score     : 64
========================================
```