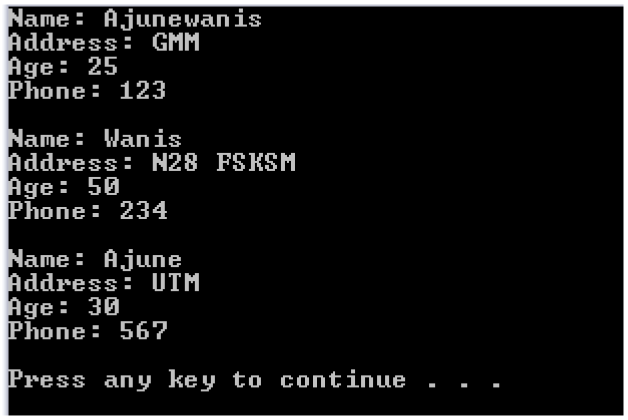
1. Create a class in your single program. Works with constructor, assessor and mutator functions.

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

// Constants
const int SIZE = 81;
// PersonalInfo class declaration
class PersonalInfo
{
private:
   char name[SIZE];
   char address[SIZE];
   int age;
   char phone[SIZE];

public:
   // Default constructor
   PersonalInfo()
   { name[0] = '\0';
     address[0] = '\0';
     age = 0;
     phone[0] = '\0';
   }

   // Overloaded Constructor
   PersonalInfo(char n[], char addr[], int a, char p[])
   {
      setName(n);
      setAddress(addr);
      setAge(a);
      setPhone(p);
   }
   // Mutator functions
   void setName(char n[])
   { strncpy(name, n, SIZE);
     name[SIZE-1] = '\0';
   }

   void setAddress(char a[])
   { strncpy(address, a, SIZE);
     address[SIZE-1] = '\0';
   }

   void setAge(int a)
   { age = a; }

   void setPhone(char p[])
   { strncpy(phone, p, SIZE);
     phone[SIZE-1] = '\0';
   }
```

```
    // Accessor functions
    const char *getName() const
    { return name; }

    const char *getAddress() const
    { return address; }

    int getAge() const
    { return age; }

    const char *getPhone() const
    { return phone; }
};
// Function prototye
void displayPersonalInfo(PersonalInfo);

int main()
{
    // Create the first instance of PersonalInfo.
    PersonalInfo me("Ajunewanis",
                    "GMM",
                    25, "123");


    PersonalInfo wanis("Wanis",
                    "N28 FSKSM",
                    50, "234");


    PersonalInfo ajune("Ajune",
                     "UTM",
                     30, "567");

    // Display the data in each object.
    displayPersonalInfo(me);
    displayPersonalInfo(wanis);
    displayPersonalInfo(ajune);
    system("pause");
    return 0;
}
void displayPersonalInfo(PersonalInfo obj)
{
    cout << "Name: " << obj.getName()
         << endl;
    cout << "Address: " << obj.getAddress()
         << endl;
    cout << "Age: " << obj.getAge()
         << endl;
    cout << "Phone: " << obj.getPhone()
         << endl << endl;
}
```

Output:



```
Name: Ajunewanis
Address: GMM
Age: 25
Phone: 123

Name: Wanis
Address: N28 FSKSM
Age: 50
Phone: 234

Name: Ajune
Address: UTM
Age: 30
Phone: 567

Press any key to continue . . .
```

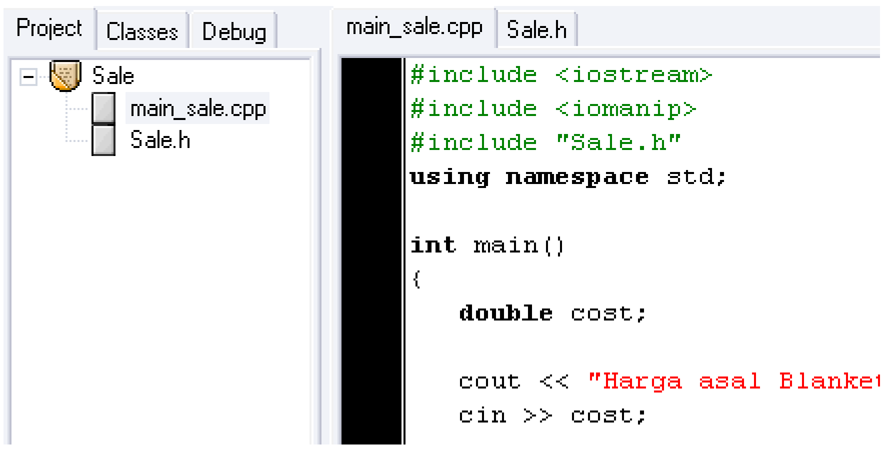
EXERCISE:

Modify this program to be more interactive which require user to define number of people, user able to enter their personal information. Prototype shown as below:

```
How many people? 2 [ENTER]
--- 1 ---
Name: Ajune [ENTER]
Address: N28 [ENTER]
Age: 20 [ENTER]
Phone: 1234455 [ENTER]
--- 2 ---
Name: Wanis [ENTER]
Address: N28 [ENTER]
Age: 20 [ENTER]
Phone: 1234455 [ENTER]
```

2. Create *external class* with header to calculate sales of an item

Your project should be sorted as below:



```
#include <iostream>
#include <iomanip>
#include "Sale.h"
using namespace std;

int main()
{
    double cost;

    cout << "Harga asal Blanket
    cin >> cost;
```

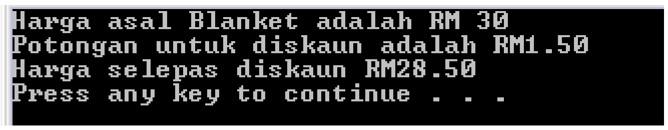
- Create main renamed as **main_sale.cpp**

```
#include <iostream>
#include <iomanip>
#include "Sale.h"
using namespace std;

int main()
{
   double cost;

   cout << "Harga asal Blanket adalah RM ";
   cin >> cost;

   Sale itemSale(cost);

   cout << fixed << showpoint << setprecision(2);

   cout << "Potongan untuk diskaun adalah RM"
        << itemSale.getDis() << endl;
   cout << "Harga selepas diskaun RM";
   cout << itemSale.getTotal() << endl;
   system ("pause");
   return 0;
}
```

- Create header file renamed with **Sale.h**

```
#ifndef SALE_H
#define SALE_H

class Sale
{
private:
   double itemCost;
   double disRate;
public:
   Sale(double cost, double discount = 0.05)
      { itemCost = cost;
        disRate = discount; }

   double getItemCost() const
      { return itemCost; }

   double getDisRate() const
      { return disRate; }

   double getDis() const
      { return (itemCost * disRate); }

   double getTotal() const
      { return (itemCost - getDis()); }
};
#endif
```

Output:



```
Harga asal Blanket adalah RM 30
Potongan untuk diskaun adalah RM1.50
Harga selepas diskaun RM28.50
Press any key to continue . . .
```

3. Create area of rectangle

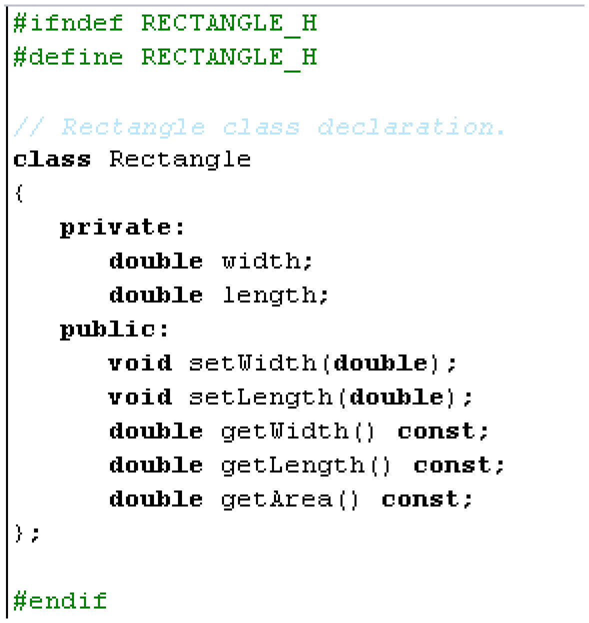
- Create main and renamed as `main.cpp`

```cpp
#include <iostream>
#include "Rectangle.h"  // Needed for Rectangle class
using namespace std;

int main()
{
   Rectangle box;      // Define an instance of the Rectangle class
   double rectWidth;   // Local variable for width
   double rectLength;  // Local variable for length

   // Get the rectangle's width and length from the user.
   cout << "The area of a rectangle: \n";
   cout << "What is the width? ";
   cin >> rectWidth;
   cout << "What is the length? ";
   cin >> rectLength;

   // Store the width and length of the rectangle
   // in the box object.
   box.setWidth(rectWidth);
   box.setLength(rectLength);

   // Display the rectangle's data.
   cout << "Here is the rectangle's data:\n";
   cout << "Width: " << box.getWidth() << endl;
   cout << "Length: " << box.getLength() << endl;
   cout << "Area: " << box.getArea() << endl;
   system ("pause");
   return 0;
}
```

- Create Rectangle cpp and renamed as `Rectangle.cpp`

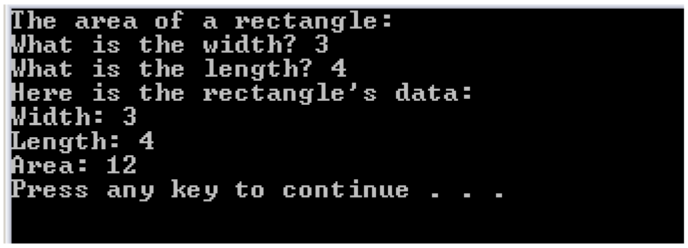
```
#include "Rectangle.h"   // Needed for the Rectangle class
#include <iostream>      // Needed for cout
#include <cstdlib>       // Needed for the exit function
using namespace std;

void Rectangle::setWidth(double w)
{
   if (w >= 0)
      width = w;
   else
   {
      cout << "Invalid width\n";
      exit(EXIT_FAILURE);
   }
}

void Rectangle::setLength(double len)
{
   if (len >= 0)
      length = len;
   else
   {
      cout << "Invalid length\n";
      exit(EXIT_FAILURE);
   }
}

double Rectangle::getWidth() const
{
   return width;
}
double Rectangle::getWidth() const
{
   return width;
}

double Rectangle::getLength() const
{
   return length;
}

double Rectangle::getArea() const
{
   return width * length;
}
```

- Create header renamed as `Rectangle.h`



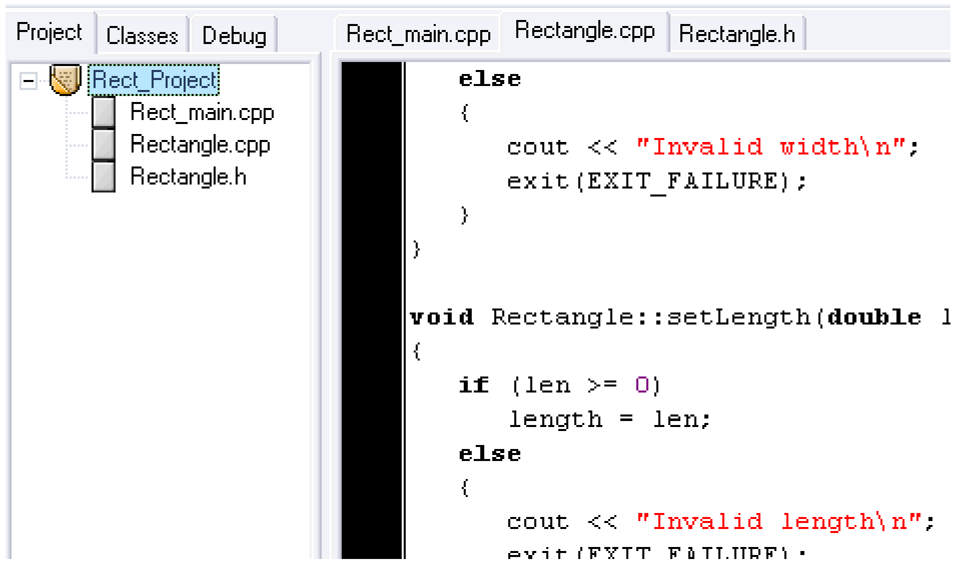
```
#ifndef RECTANGLE_H
#define RECTANGLE_H

// Rectangle class declaration.
class Rectangle
{
   private:
      double width;
      double length;
   public:
      void setWidth(double);
      void setLength(double);
      double getWidth() const;
      double getLength() const;
      double getArea() const;
};

#endif
```

Output:



```
The area of a rectangle:
What is the width? 3
What is the length? 4
Here is the rectangle's data:
Width: 3
Length: 4
Area: 12
Press any key to continue . . .
```

Remember – your project must be sorted as below



```
      else
      {
         cout << "Invalid width\n";
         exit(EXIT_FAILURE);
      }
}

void Rectangle::setLength(double l
{
      if (len >= 0)
         length = len;
      else
      {
         cout << "Invalid length\n";
         exit(EXIT_FAILURE);
```

4. How you work with **constructor** and **destructor** in a class (*refer our previous Lab Volume 1*)
   Here a simple tutorial for a class with constructor and destructor.

   - Create your main **Inventory_main.cpp**

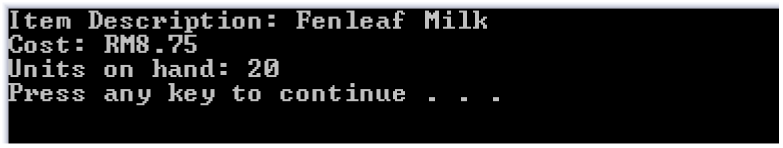
```
#include <iostream>
#include <iomanip>
#include "InventoryItem.h"
using namespace std;

int main()
{
   // Define an InventoryItem object with the following data:
   InventoryItem stock("Fenleaf Milk", 8.75, 20);

   // Set numeric output formatting.
   cout << setprecision(2) << fixed << showpoint;

   // Display the object's data.
   cout << "Item Description: " << stock.getDescription() << endl;
   cout << "Cost: RM" << stock.getCost() << endl;
   cout << "Units on hand: " << stock.getUnits() << endl;
   system("pause");
   return 0;
}
```

   - Create your header **InventoryItem.h**

```
#ifndef INVENTORYITEM_H
#define INVENTORYITEM_H
#include <cstring>   // Needed for strlen and strcpy

// InventoryItem class declaration.
class InventoryItem
{
private:
   char *description;  // The item description
   double cost;        // The item cost
   int units;          // Number of units on hand
public:
   // Constructor
   InventoryItem(char *desc, double c, int u)
      { // Allocate just enough memory for the description.
        description = new char [strlen(desc) + 1];

        // Copy the description to the allocated memory.
        strcpy(description, desc);

        // Assign values to cost and units.
        cost = c;
        units = u;}
```

```
    // Destructor
    ~InventoryItem()
       { delete [] description; }

    const char *getDescription() const
       { return description; }

    double getCost() const
       { return cost; }

    int getUnits() const
       { return units; }
};
#endif
```

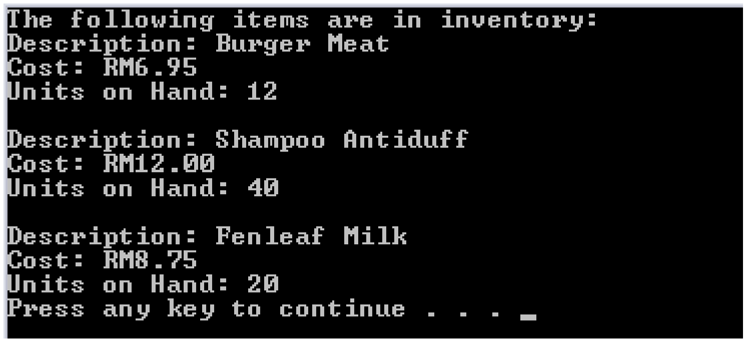
Output:



```
Item Description: Fenleaf Milk
Cost: RM8.75
Units on hand: 20
Press any key to continue . . .
```

EXERCISE

Program above shown user has defined the value of an item. Next exercise is, you need to enhance the program that displays more than 1 item (perhaps 3 items) in Inventory system. Console output as below:



```
The following items are in inventory:
Description: Burger Meat
Cost: RM6.95
Units on Hand: 12

Description: Shampoo Antiduff
Cost: RM12.00
Units on Hand: 40

Description: Fenleaf Milk
Cost: RM8.75
Units on Hand: 20
Press any key to continue . . .
```

5. Payroll system

- **Create your main for payroll system (payroll_main.cpp)**

```
#include <iostream>
#include <iomanip>
#include "Payroll.h"
using namespace std;

// Constant for number of employees
const int NUM_EMPLOYEES = 3;

int main()
{
   double hours;    // Hours worked
   double rate;     // Hourly pay rate
   int count;       // Loop counter

   // Array of Payroll objects
   Payroll employees[NUM_EMPLOYEES];

   cout << "Enter the hours worked and pay rate "
        << "for " << NUM_EMPLOYEES << " employees:\n";

   for (count = 0; count < NUM_EMPLOYEES; count++)
   {
      // Get the employee's pay rate.
      cout << "\nEmployee #" << (count+1) << " pay rate: ";
      cin >> rate;
      employees[count].setPayRate(rate);
      // Get the employee's hours worked
      cout << "Employee #" << (count+1) << " hours worked: ";
      cin >> hours;
      employees[count].setHours(hours);
   }
   // Display the total pay for each employee.
   cout << "Total pay:\n";
   cout << setprecision(2) << fixed << showpoint << right;
   for ( count = 0; count < NUM_EMPLOYEES; count++)
   {
      cout << "\tEmployee #" << (count+1) << ": ";
      cout << setw(8) << employees[count].getTotalPay() << endl;

   }
   system("pause");
   return 0;
}
```

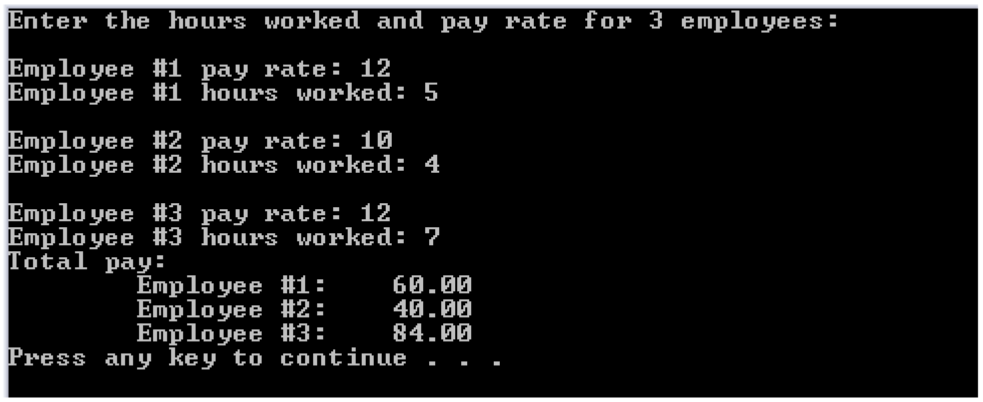
- **Create your header consists of class (*Payroll.h*)**

```
#ifndef PAYROLL_H
#define PAYROLL_H

class Payroll
{
private:
    double hours;
    double payRate;

public:
    // Constructor
    Payroll()
       { hours = 0; payRate = 0; }

    // Mutators
    void setHours(double);

    void setPayRate(double r)
        { payRate = r; }

    // Accessors
    double getHours() const
        { return hours; }

    double getPayrate() const
        { return payRate; }

    double getTotalPay() const
        { return hours * payRate; }
};
```

- **Create the implementation of your class (*payroll.cpp*)**

```
// Implementation file for the Payroll class
#include <iostream>
#include <cstdlib>
#include "Payroll.h"
using namespace std;

void Payroll::setHours(double h)
{
   if (h > 60)
   {
      // Bad number of hours.
      cout << "Invalid number of hours.\n";
      exit(EXIT_FAILURE);
   }
   else
      hours = h;
}
```

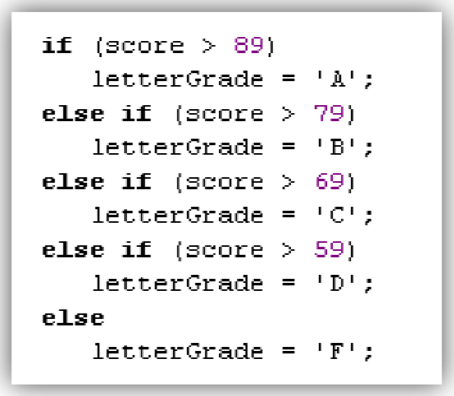
Output:



```
Enter the hours worked and pay rate for 3 employees:

Employee #1 pay rate: 12
Employee #1 hours worked: 5

Employee #2 pay rate: 10
Employee #2 hours worked: 4

Employee #3 pay rate: 12
Employee #3 hours worked: 7
Total pay:
        Employee #1:    60.00
        Employee #2:    40.00
        Employee #3:    84.00
Press any key to continue . . .
```

Assignment:

Write a program to calculate grading 3 students which sitting for their 3 subjects' test. Use assessor and mutator function. You should have *`main.cpp, student.cpp, student.h`*. Calculate and display their total score for 3 subjects' test then divide by 3 and gave them the grade they earn as stated in logic below:



```
if (score > 89)
   letterGrade = 'A';
else if (score > 79)
   letterGrade = 'B';
else if (score > 69)
   letterGrade = 'C';
else if (score > 59)
   letterGrade = 'D';
else
   letterGrade = 'F';
```

Sample prototype:

```
Enter the result of 3 test of TP2, OOP, and Web Programming for 3
students:

Student #1 TP2 : 70 [ENTER]
Student #1 OOP : 80 [ENTER]
Student #1 Web Programming : 60 [ENTER]
Total Score : 210
Average : 70
Grade: B
.
.
.
```