

Student Name \_\_\_\_\_ Section \_\_\_\_\_

Instructor \_\_\_\_\_ Due Date \_\_\_\_\_

Project	1	2	3	4	5	6	TOTAL
Maximum Points	10 points	10 points	20 points	20 points	20 points	20 points	100 points
Your Score							

## PROJECT ONE

**Objective** To type, compile and execute a C++ computer program.

### PROJECT DESCRIPTION

Use the C++ program, which follows, to compute various values.

#### Information About This Project

This project introduces you to the C++ computer programming language. This high-level programming language is considered a useful tool to introduce us to the subject of computer programming.

#### Steps To Complete This Project

##### STEP 1 Open Visual Studio 2005

Start Microsoft Visual Studio 2005 on your computer. The typical path to open this application is click the **Start** button on your Windows **Desktop**, click **All Programs**, point to **Microsoft Visual Studio 2005**, point to and click on **Microsoft Visual Studio 2005**.

##### STEP 2 Open a New Visual C++ 2005 Project

Create a new Visual Studio project as follows: Click **File** on the main menu bar, point to and click on **New Project...**. When the **New Project** dialog box opens, select and fill the **New Project** dialog box with these entries.

- **Project Types** select Visual C++ ( which appears under " Other Languages ")
- **Templates** select Win32 Console Application
- **Name** Type over <Enter name> in the Name field of the dialog box and key - in **Project One**
- **Location** choose the location ( drive / folder ) you wish to save your files

Click **OK** to close the **New Project** dialog box. MS Visual Studio will now take you to the **Win32 Application Wizard**. This Wizard generates a **Win32 Application** project. Within the Wizard, select the link to:

#### Application Settings

Then under **Application type**, choose the option:

- Console application

For the **Additional options**, click the following check box:

- Empty project

Click the **Finish** button to close the Wizard.

Visual Studio 2005 now creates a solution containing **Project One**.

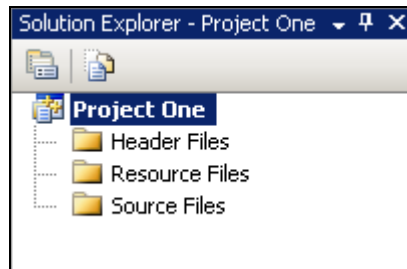
Student Name \_\_\_\_\_

Section \_\_\_\_\_

## PROJECT ONE

**STEP 3 Add a C ++ Source File to Your Project**

On the main menu, click **View** and select **Solution Explorer**. Navigate to the Solution Explorer and select ( left - click on ) **Project One**.



On the main menu, click **Project** and select **Add New Item**. When the **Add New Item** dialog box appears, ensure that the Visual C ++ category is selected and then choose the template C ++ File ( .cpp ) to create a C ++ source file. Type over <Enter name> in the Name field of the dialog box and type **lab1prj1** as the name of the file.

Click **Add** to close the **Add New Item** dialog box.

**STEP 4 Add Code to Your C ++ Source File**

MS Visual Studio now takes you to the **Code** window of the **lab1prj1.cpp** file. With your cursor located within the Code window of the **.cpp** file, type the following code located in **Figure 1**. Use your own name in place of Sammy Student.

**Figure 1 Source Code**

```
#include <iostream>
using namespace std;
void main()
{
    //Sammy Student, Programmer
    //Program that computes the area of a triangle
    double area, base, height;
    cout << "\nplease enter the base ";
    cin >> base;
    cout << "\nplease enter the height ";
    cin >> height;
    area = 0.50 * base * height;
    cout << "\nthe area of the triangle is " << area;
    cout << "\n\n";
    cout << "press [Enter] to close this window\n";
    cin.get ();
}
```

**STEP 5 Run Your Visual Studio Application and Observe the Output**

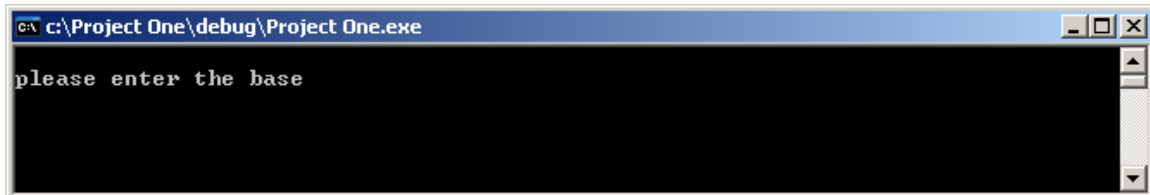
After coding your application, build your project. Click **Build** on the main menu and select **Build Project One**.

Student Name \_\_\_\_\_

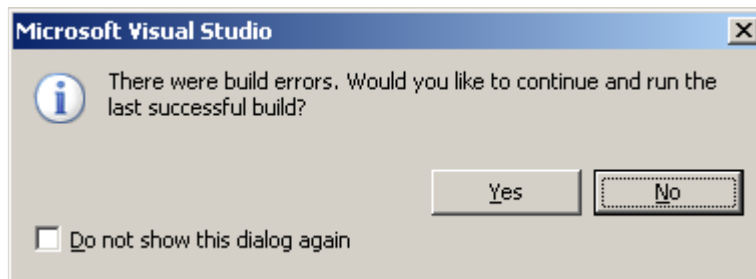
Section \_\_\_\_\_

**PROJECT ONE**

Now run the application to test your project. Either click your **F5** keyboard function key or click **Debug** on the main menu and select **Start Debugging**. Visual Studio now compiles and runs your program and takes you to an output screen. If your program has compiled without syntax errors, the following output screen will appear.



If your program has compiled with syntax errors, the following message box may appear.



If the above message appears, click **No** and scroll through the **Error List** window to view any error messages. Read the error messages and make any necessary corrections by comparing your code to that shown on the prior page. Then re - run your program by pressing **F5**.

With your program now error free, close the output window.

Now re - run your program and bypass the "Debugger" and execute your program without debugging by pressing **Ctrl** + **F5**.

With your output screen visible, follow the prompts and enter the following sample program run information.

Triangle Data	
base	5
height	10

**STEP 6 Check Your Output**

Verify the logic of your program by checking your output with the sample run shown below. Your output screen will appear similar to the one below.

**Program Output**

```
please enter the base    5
please enter the height 10
the area of the triangle is 25
```

Student Name \_\_\_\_\_

Section \_\_\_\_\_

## PROJECT ONE

After verifying the output, press your spacebar to return to Visual C++.

Your program has just computed the area of a triangle whose base length is 5 and whose height length is 10. The area of any triangle is given by the formula

$A = \frac{1}{2} b h$  where  $b$  represents the base of the triangle and  $h$  represents the height of the triangle.

**STEP 7** **Modify Your Program**

Next, modify your program such that it computes the area of a trapezoid instead of the area of a triangle.

The area  $A$  of a trapezoid is given by the formula  $A = \frac{1}{2} h (b_1 + b_2)$  where  $h$  represents the height of the trapezoid and  $b_1, b_2$  are the lengths of the bases of the trapezoid.

To modify your program, first change your program comment statement to indicate that your program will compute the area of a trapezoid, i.e. use:

```
//Program that computes the area of a trapezoid
```

Next, modify your variable declaration statement to include both bases.

```
double area, base1, base2, height;
```

Also, modify your input statements to include both bases.

```
cout << "\nplease enter the base 1  ";
cin >> base1;
cout << "\nplease enter the base 2  ";
cin >> base2;
cout << "\nplease enter the height  ";
cin >> height;
```

Change your assignment statement such that it now computes the area of a trapezoid, as shown below.

```
area = 0.50 * height * (base1 + base2);
```

Finally, change your output statement to read as follows:

```
cout << "\nthe area of the trapezoid is " << area;
```

**STEP 8** **Re - Compile Your Program**

Click **Build** on the main menu and select **Rebuild Project One**.

Next re - compile your program by pressing **F5**. Check for any syntax errors. If your program has compiled with syntax errors, scroll through the **Error List** window to view any error messages. Make any necessary corrections. Then recompile your program.

If your program is running correctly, you are taken to the black output screen. Close the screen. Then run your program, without debugging by pressing **Ctrl** + **F5**.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT ONE****STEP 9**      **Run Your Modified Program**

While your program is running, determine the area, in square inches, of a trapezoid whose height is 5.75 inches and whose bases  $b_1$  and  $b_2$  are 10.25 and 8.75 inches, respectively. The correct answer that you should arrive at is 54.625 .

**STEP 10**      **Print Your Modified Program Code**

Once you have determine that your modified program is correctly computing the area of a trapezoid, print your program code and submit the output for credit. While you are in the **Code** window, use the **File** menu and then **Print...** to print the program code.

Make sure that your name appears in the heading portion of the code.

**STEP 11**      **Close Your Visual Studio Application**

To close your file, click **File** and select **Close** . Then close your project by clicking **File** and selecting **Close Project** .

You can now create a new project by clicking **File** on the main menu bar, pointing to and choosing **New Project ...** .

Student Name \_\_\_\_\_ Section \_\_\_\_\_

**PROJECT TWO****Objective** To key - in, compile and execute a cash register simulation computer program.**PROJECT DESCRIPTION**

Type, compile and execute the price / quantity program shown in **Figure 1** and then modify the program to simulate a simple cash register.

**Information About this Project**

This project introduces you a simple program that requests the price and quantity of a particular item, multiplies these two variables to calculate the subtotal and then adds the tax to obtain the total amount due.

**Steps To Complete This Project****STEP 1 Open MS Visual C++ 2005 and Type the Program Code**

Launch MS Visual Studio 2005 on your computer, if it is not already open. Start a new project ( Visual C++ Project / Win32 Console Application ) by clicking **New Project...** from the **File** menu. Name your project as: **Project Two**

**STEP 2 Add a C++ Source File to Your Project**

On the main menu, click **View** and select **Solution Explorer**. Navigate to the Solution Explorer and select your project, in this case **Project Two**.

On the main menu, click **Project** and select **Add New Item**. When the **Add New Item** dialog box appears, ensure that the Visual C++ category is selected and then choose the template C++ File ( **.cpp** ) to create a C++ source file. Type over <Enter name> in the Name field of the dialog box and type **lab1prj2** as the name of the file.

Click **Add** to close the **Add New Item** dialog box.

**STEP 3 Add Code to Your C++ Source File**

MS Visual Studio now takes you to the **Code** window of the **lab1prj2.cpp** file. With your cursor located within the Code window of the **.cpp** file, type the following code located in **Figure 1**. Use your own name in place of Sammy Student.

**STEP 4 Compile Your Program**

After coding your application, test it by running your project. Either click your **F5** function key or click **Debug** on the main menu and select the **Start Debugging** option.

If your program is running correctly, you are taken to the black output screen. Close this screen and proceed to the next step.

If your program has compiled with syntax errors, scroll through the **Error List** window to view any error messages. Read the error messages and make any necessary corrections by comparing your code to that shown on the prior page. Then re - run your program by pressing **F5**. Close the black output screen and proceed to the next step.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

## PROJECT TWO

**Figure 1 Program Code for Cash Register Program**

```

/*****\
*Program to demonstrate a
*simple cash register program
*written by Sammy Student
*today's date
\*****/

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

void main()
{
    int quantity;
    double price, subtotal;
    double tax, total;

    cout << "please enter the item price:   ";
    cin >> price;
    cout << "please enter the item quantity: ";
    cin >> quantity;

    cout.setf(ios::fixed);
    cout.precision(2);

    subtotal = price * quantity;
    cout << "\nsubtotal:  \t$" << setw(10) << subtotal << endl;

    tax = subtotal * 0.05;
    cout << "\ntax:      \t$" << setw(10) << tax << endl;

    total = tax + subtotal;
    cout << "\ntotal:   \t$" << setw(10) << total << endl;

    cout << "\n\n";
    cout << "press [Enter] to close this window\n";
    cin.get();
}

```

**STEP 5****Run Your Program**

After your program has successfully compiled, execute ( run ) your program without debugging by pressing **Ctrl** + **F5**.

While your program is running, enter the following sample program run information, when prompted.

Item Data	
price	\$ 5.50
Quantity	100

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT TWO****STEP 6 Check Your Output**

Verify the logic of your program by checking your output with the sample run shown below. Your output screen will appear similar to the one below.

**Program Output**

```
please enter the item price:      5.50
please enter the item quantity: 100

subtotal:      $      550.00
tax:           $       27.50
total:        $      577.50
```

Your program has just computed the total amount due when a quantity of 100 items is ordered, each priced at \$ 5.50 per item. The tax rate of the order is 5%. After you have examined your output, press **Enter** to return to Visual C++.

**STEP 7 Modify Your Program**

Next, modify your program such that it not only computes the total amount due but also inputs the amount of cash received from the customer on the purchase order and calculates the amount of change. Assume that the customer always gives the exact amount or an amount greater than the actual amount due, thereby making the change due always a positive amount.

**STEP 8 Re - Compile Your Program**

Click **Build** on the main menu and select **Rebuild Project Two**.

Click **Debug** and then choose **Start Debugging**.

If your program has compiled without syntax errors, proceed to the next step, otherwise scroll through the **Error List** window to view any error messages. Make any necessary corrections. Then recompile your program.

**STEP 9 Run Your Modified Program**

After your program has successfully compiled, execute ( run ) your program by pressing **Ctrl** + **F5**.

While your program is running, determine the amount of change when a customer orders a quantity of 100 items, each priced at \$ 5.50 per item ( the tax rate of the order is still 5% ), and gives the clerk six one - hundred dollar bills.

**STEP 10 Print Your Modified Program Code**

Once you have determined that your modified program is correctly computing the amount of change, print your program code and submit the output for credit. Make sure that your name appears in the heading portion of the code.

**STEP 11 Close Your Visual Studio Application**

To close your file, click **File** and select **Close**. Then close your project by clicking **File** and selecting **Close Project**.

You can now create a new project by clicking **File** on the main menu bar, pointing to and choosing **New Project ...**.

Student Name \_\_\_\_\_ Section \_\_\_\_\_

**PROJECT THREE****Objective** To type, compile and execute a program that computes simple interest.**PROJECT DESCRIPTION**

This project has you running a program that computes simple interest.

**Information About This Project**

According to the Mathematics of Finance, simple interest is defined as interest, which is based upon the original principal only. The formula to compute simple interest is given by

$$\text{Interest Earned} = \text{Principal} \times \text{Rate} \times \text{Time}$$

or algebraically,

$$I = P \cdot R \cdot T$$

where  $I$  is the Interest earnings,  $P$  is the Principal or the amount invested,  $R$  is the annual interest Rate and  $T$  is the Time that the money is kept on deposit, expressed in terms of years

Note: that if  $R$  is given as an annual interest rate, then the time  $T$  must be expressed in years.

Under the rules of simple interest, the amount in the account  $A$  after a principal  $P$  has been invested for a period of  $T$  years is given as:

$$A = P + I$$

Since  $I = P \cdot R \cdot T$  then the above formula can also be expressed as:

$$A = P + P \cdot R \cdot T$$

or in the familiar factored form:

$$A = P (1 + R \cdot T)$$

This last formula is useful to determine one of the variables  $A$ ,  $P$ ,  $R$  or  $T$  when the remaining three variables are known.

**Steps To Complete This Project****STEP 1 Open MS Visual C++ 2005 and Type the Program Code**

Launch MS Visual Studio 2005 on your computer, if it is not already open. Start a new project ( Visual C++ Project / Win32 Console Application ) by clicking **New Project...** from the **File** menu. Name your project as: **Simple\_Interest**

**STEP 2 Add a C++ Source File to Your Project**

On the main menu, click **View** and select **Solution Explorer**. Navigate to the Solution Explorer and select **Simple\_Interest**.

On the main menu, click **Project** and select **Add New Item**. When the **Add New Item** dialog box appears, ensure that the Visual C++ category is selected and then choose the template C++ File ( **.cpp** ) to create a C++ source file. Type over <Enter name> in the Name field of the dialog box and type **simple** as the name of the file.

Click **Add** to close the **Add New Item** dialog box.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

## PROJECT THREE

**STEP 3 Add Code to Your C++ Source File**

MS Visual Studio now takes you to the **Code** window of the **simple.cpp** file. With your cursor located within the Code window of the **.cpp** file, type the following code located in **Figure 1**. Use your own name in place of Sammy Student.

**Figure 1 Simple Interest Program Code**

```
#include <iostream>
#include <iomanip>
using namespace std;
//Sammy Student, Programmer

int main()
{
    //declare the variables
    double amount, earnings, principal, rate, time;
    char answer;

    do
    {
        //prompt the user for data and read the values
        cout << "Compute simple interest and total amount\n" << endl;
        cout << "Enter the deposit amount (no $ or comma)" << endl;
        cin >> principal;
        cout << "Enter the annual interest rate as a decimal" << endl;
        cin >> rate;
        cout << "Enter the time of deposit" << endl;
        cin >> time;

        //perform the computations
        earnings = principal * rate * time;
        amount = principal + earnings;
        //display the computations
        cout << "Amount in Account = "
             << setiosflags(ios::fixed | ios::showpoint)
             << setprecision(2) << amount << endl;
        cout << "Simple Interest " << setprecision(2) << earnings << endl;
        cout << "Would you like to run another calculation?" << endl;
        cout << "(Y/N)?\n";
        cin >> answer;
    }
    while(answer == 'y' || answer == 'Y');
    return 0;
}
```

**STEP 4 Compile and Test your Program**

Use  +  to compile your program.

Test your program by trying the input information, which follows, and then observe if the correct output is obtained.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

## PROJECT THREE

Principal	\$ 30,000.00	input
Rate	6.5 %	
Time	5 years	
Interest Earnings	\$ 9,750.00	output
Amount in Account	\$ 39,750.00	

**STEP 5**      **Run Your Program for Various Scenarios**

Use your program to compute the Simple Interest and the Amount In Account for each of the cases shown within **Figure 2**, which follows.

Type the individual results into a new MS Word or MS WordPad document. Label each of the case results within the document, similar to the style shown within **Figure 2**, which follows.

Submit the document for credit.

**STEP 6**      **Print Your Program Code**

Finally, paste your program code into your document and submit it for credit.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

## PROJECT THREE

**Figure 2 Simple Interest Scenarios**

**Instructions** Use your program to compute the simple interest earnings and the accumulated amount in the account for each of the cases below. Copy and label the results into a new MS Word or MS WordPad document.

(a) *Principal* \$ 3,000.00  
*Interest Rate* 6.5 %  
*Time* 4 years  
*Interest Earnings* \_\_\_\_\_  
*Amount in Account* \_\_\_\_\_

(b) *Principal* \$ 4,550.00  
*Interest Rate* 1.75 %  
*Time* 5 years  
*Interest Earnings* \_\_\_\_\_  
*Amount in Account* \_\_\_\_\_

(c) *Principal* \$ 650.50  
*Interest Rate* 5.25 %  
*Time* 1 year  
*Interest Earnings* \_\_\_\_\_  
*Amount in Account* \_\_\_\_\_

(d) *Principal* \$ 25,600.00  
*Interest Rate* 6.30 %  
*Time* 10 years  
*Interest Earnings* \_\_\_\_\_  
*Amount in Account* \_\_\_\_\_

(e) *Principal* \$ 9,286.42  
*Interest Rate* 2.75 %  
*Time* 9 months  
*Interest Earnings* \_\_\_\_\_  
*Amount in Account* \_\_\_\_\_

(f) How much interest will Jane Windsor earn on an \$ 800.00 investment at 8.50 % simple interest for 2 years? What will be the amount in her account after 2 years?  
*Interest Earnings* \_\_\_\_\_  
*Amount in Account* \_\_\_\_\_

Student Name \_\_\_\_\_ Section \_\_\_\_\_

**PROJECT FOUR****Objective** To write computer code that will compute and report generate data.**PROJECT DESCRIPTION**

Write, compile and execute a program that computes both an employee's gross and net pay and then generates a report of the employee's payroll data. Your program is to be written such that it generates a payroll report similar to the sample report shown below.

**Sample Payroll Report**

Employee Number	25443
Number of Hours Worked This Week	38
Hourly Wage	\$ 8.75
Gross Pay	\$ 332.50
Total Tax Deductions	\$ 59.86
Net Pay	\$ 272.64

All employees are subject to a flat 15 % federal tax rate on the gross pay and a flat 3 % state tax rate on the gross pay.

Assume that the employees are not paid any overtime pay.

**Information About this Project**

To compute the employee's gross pay, use the following formula:

$$\text{Gross Pay} = \text{Hourly Wage} \times \text{Number of Hours}$$

To compute the employee's total taxes, use the following formula:

$$\text{Total Taxes} = \text{Federal Tax} + \text{State Tax}$$

To compute the employee's net pay, use the following formula:

$$\text{Net Pay} = \text{Gross Pay} - \text{Total Taxes}$$

**Steps To Complete This Project****STEP 1 Open MS Visual C++ 2005 and Type the Program Code**

Launch MS Visual Studio 2005 on your computer, if it is not already open. Start a new project ( Visual C++ Project / Win32 Console Application ) by clicking **New Project...** from the **File** menu. Name your project as: **Project Four**

**STEP 2 Add a C++ Source File to Your Project**

On the main menu, click **View** and select **Solution Explorer**. Navigate to the Solution Explorer and select **Project Four**.

On the main menu, click **Project** and select **Add New Item**. When the **Add New Item** dialog box appears, ensure that the Visual C++ category is selected and then choose the template C++ File (.cpp) to create a C++ source file. Type over <Enter name> in the Name field of the dialog box and type **lab1prj4** as the name of the file. Click **Add** to close the **Add New Item** dialog box.

**STEP 3 Write the Program Code**

Next, write the program code corresponding to the **Project Description** given above.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT FOUR**

Here are some hints and direction to write your code:

- Include the typical pre - processing directives and open a `main()` method.
- Declare an integer variable named `empNumber` that will store the employee's ID number. Initialize the variable to the number 0.
- Declare a `double` type variable named `numHours` that will store the employee's number of labor hours. Initialize the variable to 0.
- Declare a `double` type variable named `dblWage` that will store the employee's hourly wage. Initialize the variable to 0.
- Declare a `double` type variable named `grossPay` that will store the employee's gross pay. Initialize the variable to 0.
- Declare a `double` type variable named `fedTaxes` that will store the employee's federal tax deductions. Initialize the variable to 0.
- Declare a `double` type variable named `stateTaxes` that will store the employee's state tax deductions. Initialize the variable to 0.
- Declare a `double` type variable named `totalTaxes` that will store the employee's total tax deductions. Initialize the variable to 0.
- Declare a `double` type variable named `net_pay` that will store the employee's net pay. Initialize the variable to 0.
- Prompt the user for the employee's ID number and store the user's response in the variable named `empNumber`.
- Prompt the user for the employee's number of labor hours and store the user's response in the variable named `numHours`.
- Prompt the user for the employee's hourly wage and store the user's response in the variable named `dblWage`.
- Compute the employee's gross pay by multiplying variable `numHours` by variable `dblWage` and store the product in variable `grossPay`.
- Compute the employee's federal tax by multiplying variable `grossPay` by the applicable rate and store the product in variable `fedTaxes`.
- Compute the employee's state tax by multiplying variable `grossPay` by the applicable rate and store the product in variable `stateTaxes`.
- Compute the employee's total tax by adding variable `fedTaxes` to variable `stateTaxes` and store the sum in variable `totalTaxes`.
- Compute the employee's next pay by subtracting variable `totalTaxes` from variable `grossPay` and store the sum in variable `net_pay`.
- Display each of these labels and their corresponding current variable values to the user: employee's ID number, the number of labor hours, the hourly wage, the gross pay, the total tax deductions and the net pay.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT FOUR****STEP 4 Compile and Test Your Program**

After you write your program source code, click **F5** to compile your program.

If your program has successfully compiled, execute ( run ) your program.

While your program is running, enter the sample input values shown within the **Sample Payroll Report** that appears within the **Project Description** portion of this individual project. Observe your program output. It is to be similar to the sample payroll report. If it is, keep your output screen open and proceed to the next step. Otherwise, make any necessary adjustments / corrections to your source code and run your program again.

**STEP 5 Print a Screen Snapshot of Your Output Screen**

With your output screen open still open and visible in your program run mode, press and hold your keyboard **Alt** key and then press your **Print Screen** key. Then, leaving your program open, paste the screen snapshot into a word processor such as MS Word by opening MS Word and, in a new document, pressing and holding your **Ctrl** key and then pressing your **V** key.

Now, below your screen snapshot in your word processing document, press your **Enter** key a few times and then type your name, the course number and date. Finally, print a copy of your Word document and submit it for credit. Close your word processor and when you return to your program's output screen, press your spacebar to return to Visual C++.

**STEP 6 Print Your Program Code**

Once you have determined that your modified program is correctly displaying the required report, print your program code and submit the output for credit. Make sure that your name appears in the heading portion of the code.

**STEP 7 Close Your Visual Studio Application**

To close your file, click **File** and select **Close**. Then close your project by clicking **File** and selecting **Close Project**.

You can now create a new project by clicking **File** on the main menu bar, pointing to and choosing **New Project ...**.

Student Name \_\_\_\_\_ Section \_\_\_\_\_

**PROJECT FIVE****Objective** To write a program that calculates the distance an object travels in free fall.**PROJECT DESCRIPTION**

Write a program that prompts the user to enter a value for the time  $t$  and then computes the distance  $s$  that an object under free fall will travel in that time.

**Information About this Project**

According to Physics, the formula for the distance a body travels in free fall is given by the formula.

$$s = \frac{1}{2} g t^2$$

where

$s$  is the distance traveled, in feet

$t$  is the time in seconds to travel the distance  $s$

$g$  is a constant value called the gravitational constant of acceleration and is equal to 32.2 feet per second<sup>2</sup>

**Steps To Complete This Project****STEP 1 Open a New MS Visual C++ 2005 Project**

Launch MS Visual Studio 2005 on your computer, if it is not already open. Start a new project ( Visual C++ Project / Win32 Console Application ) by clicking **New Project...** from the **File** menu. Name your project as: **Project Five**

**STEP 2 Add a C++ Source File to Your Project**

On the main menu, click **View** and select **Solution Explorer**. Navigate to the Solution Explorer and select **Project Five**.

On the main menu, click **Project** and select **Add New Item**. When the **Add New Item** dialog box appears, ensure that the Visual C++ category is selected and then choose the template C++ File ( **.cpp** ) to create a C++ source file. Type over <Enter name> in the Name field of the dialog box and type **lab1prj5** as the name of the file. Click **Add** to close the **Add New Item** dialog box.

**STEP 3 Write the Program Code**

Review your **Project Description** above for the coding requirements of this project.

Here are some hints and direction to write your code:

- Include the following pre - processing directives:

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

- Include comment statements to remark upon the comments of the program and to identify you as the programmer.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT FIVE**

- Open the scope of the `main()` method.  

```
void main()
{
```
- Declare and initialize all the variables.  

```
double distance = 0;
const double g_force = 32.2;
double time = 0;
```
- Prompt the user for the required information and update the variable(s).  

```
cout << "please enter is the time (in seconds) ";
cin >> time;
```
- Perform any required computations.  

```
distance = 0.50 * g_force * time * time;
```
- Display the required output to the user.  

```
cout << " the distance traveled in " << time
<< " seconds is "
<< setiosflags(ios::fixed | ios::showpoint)
<< setprecision(3) << distance << endl;
```
- Close the scope of the `main()` method.  

```
}
```

**STEP 4 Compile and Test Your Program**Click  to compile your program.**STEP 5 Run Your Program**

After your program has successfully compiled, execute ( run ) your program. While your program is running, enter the following sample program run information, when prompted.

Test your program using the following data. By entering a time value from the list below, your program should correctly compute the corresponding distance that the object has fallen in that period of time.

Time ( in seconds )	Distance ( in feet )
1.00	16.100
6.50	680.225
12.40	2475.536

Keep your output screen open, which shows the correct distance traveled for a particular time, from the above chart. Proceed to the next step.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT FIVE****STEP 6 Print a Screen Snapshot of Your Output Screen**

With your output screen open still open and visible in your program run mode, press and hold your keyboard **[Alt]** key and then press your **[Print Screen]** key. Then, leaving your program open, paste the screen snapshot into a word processor such as MS Word by opening MS Word and, in a new document, pressing and holding your **[Ctrl]** key and then pressing your **[V]** key.

Now, below your screen snapshot in your word processing document, press your **[Enter]** key a few times and then type your name, the course number and date. Finally, print a copy of your Word document and submit it for credit. Close your word processor and when you return to your program's output screen, press your spacebar to return to Visual C++.

**STEP 7 Print Your Program Code**

Once you have determine that your modified program is correctly computing the distance traveled under free fall, print your program code and submit the output for credit. Make sure that your name appears in the heading portion of the code.

**STEP 8 Close Your Visual Studio Application**

To close your file, click **[File]** and select **[Close]**. Then close your project by clicking **[File]** and selecting **[Close Project]**.

You can now create a new project by clicking **[File]** on the main menu bar, pointing to and choosing **[New Project ...]**.

Student Name \_\_\_\_\_ Section \_\_\_\_\_

**PROJECT SIX****Objective** To write a program illustrates sequential program control.**PROJECT DESCRIPTION**

Write, compile and run one complete program that performs the following multiple instructions / computations in a sequential ( step - by - step ) order. Each of the individual programming steps below are to be accomplished by a single line of program code.

Include your name, lab project / number and other course information in the header portion of your program.

**Program Statements**

- Declare `num1` as an integer type variable and assign `num1` an initial value of `6`.
- On a new line of output, display to the program user the name and the initial value of variable `num1`.
- Declare `num2` as an integer type variable and assign `num2` an initial value of twice the value of variable `num1`.
- On a new line of output, display to the program user the name and the initial value of variable `num2`.
- Declare `num3` as an integer type variable and assign `num3` to be the difference between the current values of `num1` and `num2`.
- On a new line of output, display to the program user the name and the initial value of variable `num3`.
- Increment `num1` by adding the number `3` to its current value.
- Decrease `num2` to one - fourth of its current value.
- On a new line of output, display to the program user the product of `num1` and `num2`.
- On a new line of output, display to the program user the product of `num2` and `num3`.
- On a new line, display to the program user the quotient when `num1` is divided by `num3`.
- Change the current value of `num3` such that it will now be assigned twice the value of the sum of `num1` and `num2`.
- On a new line of output, display to the program user the word `total` followed by one space followed by the sum of the current values of `num1`, `num2` and `num3`.

Student Name \_\_\_\_\_

Section \_\_\_\_\_

**PROJECT SIX*****Information About this Project***

Sequential program control is the most basic of the three program controls, namely: sequential, selection and looping. This type of program control has computer programmers write lines of code such that tasks are performed in separate line - by - line instructions.

***Steps To Complete This Project*****STEP 1      *Open MS Visual C++ 2005 and Write the Program Code***

Open Visual C++ on your computer. Write the program code that will allow the user to enter the necessary input items and then use these items to compute the required output value(s). Save your file as: **lab1prj6.cpp**

**STEP 2      *Compile and Run Your Program***

Build, compile and run your program. Test the operation of your program.

**STEP 3      *Print your Program Code and Your Run Time Output***

When completed, print your program source code as well as a screen snapshot of your program output. Attach the hardcopies to your lab cover sheet for credit.