

Student Name _____ Section _____
 Instructor _____ Due Date _____

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

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 the UNIX emacs Text Editor and Type the Program Code**

Open a new UNIX session and, at the UNIX command prompt, type the following command to invoke the emacs editor.

```
emacs lab1prj1.cpp
```

When the emacs editor opens, type the program code shown in **Figure 1**, which follows, exactly as it appears, except substitute 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();
}
```

Student Name _____

Section _____

PROJECT ONE**STEP 2 Compile the Program**

After you have typed the given program, click `Tools` on the emacs menu and then select `Compile`. The cursor will shift to the bottom of the screen and the following will be displayed:

```
Compile command: make -k
```

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj1.cpp -o lab1prj1.cpp
```

Press `Enter` to have the command accepted. If the program compiles correctly, you will see a message such as

```
Compilation finished at Mon June 13 9:15:33
```

If you do not have any errors, proceed to the next step, otherwise read the error messages and make any necessary corrections by comparing your code to that shown on the prior page. Then recompile your program.

STEP 3 Run the Program

Close emacs and return to the UNIX command prompt.

At the UNIX prompt, type the following command and press `Enter` to execute the program.

```
lab1prj1
```

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

Triangle Data	
base	5
height	10

STEP 4 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
```

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.

After verifying the output, type the following at the UNIX prompt to return to your source code and the emacs editor.

```
emacs lab1prj1.cpp
```

Student Name _____

Section _____

PROJECT ONE

STEP 5 **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 6 **Re - Compile Your Program**

After you have typed the given program, click **Tools** on the emacs menu and then select **Compile**. The cursor will shift to the bottom of the screen and the following will be displayed:

```
Compile command: make -k
```

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj1.cpp -o lab1prj1.cpp
```

Press **Enter** to have the command accepted. If the program compiles correctly, you will see a message such as

```
Compilation finished at Mon June 13 9:15:33
```

If you do not have any errors, proceed to the next step, otherwise read the error messages and make any necessary corrections by comparing your code to that shown on the prior page. Then recompile your program.

Student Name _____

Section _____

PROJECT ONE**STEP 7** **Run Your Modified Program**

Close `emacs` and return to the UNIX command prompt.

At the UNIX prompt, type the following command and press `Enter` to execute the program.

```
lab1prj1
```

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 feet, respectively. The correct answer that you should arrive at is 54.625 .

STEP 8 **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.

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

STEP 9 **Close `emacs` and Return to the UNIX Prompt**

Before proceeding to the next program, close your `emacs` screen to return to the UNIX prompt. You are ready to proceed to the next 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 the UNIX emacs Text Editor and Type the Program Code**

Open a new UNIX session and, at the UNIX command prompt, type the following command to invoke the emacs editor.

```
emacs lab1prj2.cpp
```

When the emacs editor opens, type the program code shown in **Figure 1**, which follows, exactly as it appears, except substitute your own name in place of Sammy Student.

STEP 2 **Compile the Program**

After you have typed the given program, click **Tools** on the emacs menu and then select **Compile**. The cursor will shift to the bottom of the screen and the following will be displayed:

```
Compile command: make -k
```

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj2.cpp -o lab1prj2.cpp
```

Press **Enter** to have the command accepted. If the program compiles correctly, you will see a message such as

```
Compilation finished at Mon June 13 9:15:33
```

If you do not have any errors, proceed to the next step, otherwise read the error messages and make any necessary corrections by comparing your code to that shown on the prior page. Then recompile your program.

STEP 3 **Run the Program**

Close emacs and return to the UNIX command prompt.

At the UNIX prompt, type the following command and press **Enter** to execute the program.

```
lab1prj2
```

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();
}

```

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 4 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%.

STEP 5 Modify Your Program

After verifying the output, type the following at the UNIX prompt to return to your source code and the emacs editor.

```
emacs lab1prj2.cpp
```

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 6 Run Your Modified Program

Close emacs and return to the UNIX command prompt.

At the UNIX prompt, type the following command and press to execute the program.

```
lab1prj2
```

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 7 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 8 Close emacs and Return to the UNIX Prompt

Before proceeding to the next program, close your emacs screen to return to the UNIX prompt. You are ready to proceed to the next 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 the UNIX emacs Text Editor and Type the Program Code**

Open a new UNIX session and, at the UNIX command prompt, type the following command to invoke the emacs editor.

```
emacs lab1prj3.cpp
```

When the emacs editor opens, type the program code shown in **Figure 1**, which follows, exactly as it appears, except substitute your own name in place of Sammy Student.

STEP 2 **Compile the Program**

After you have typed the given program, click **Tools** on the emacs menu and then select **Compile**. The cursor will shift to the bottom of the screen and the following will be displayed:

```
Compile command: make -k
```

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj3.cpp -o lab1prj3.cpp
```

Student Name _____ Section _____

PROJECT THREE

Press to have the command accepted. If the program compiles correctly, you will see a message such as

```
Compilation finished at Mon June 13 9:15:33
```

If you do not have any errors, proceed to the next step, otherwise read the error messages and make any necessary corrections by comparing your code to that shown below. Then recompile your program.

Figure 1 Simple Interest Program Code

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

void main()
{
    double amount, earnings, principal, rate, time;
    char answer;

    do
    {
        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;
        earnings = principal * rate * time;
        amount = principal + earnings;
        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');
}
```

STEP 3**Run the Program**

Close emacs and return to the UNIX command prompt.

At the UNIX prompt, type the following command and press to execute the program.

```
lab1prj3
```

Student Name _____

Section _____

PROJECT THREE

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

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

STEP 4 **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 5 **Print Your Program Code**

Finally, print your program code and attach it to this lab packet 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) | <i>Principal</i> | \$ 3,000.00 |
| | <i>Interest Rate</i> | 6.5 % |
| | <i>Time</i> | 4 years |
| | <i>Interest Earnings</i> | _____ |
| | <i>Amount in Account</i> | _____ |
| | | |
| (b) | <i>Principal</i> | \$ 4,550.00 |
| | <i>Interest Rate</i> | 1.75 % |
| | <i>Time</i> | 5 years |
| | <i>Interest Earnings</i> | _____ |
| | <i>Amount in Account</i> | _____ |
| | | |
| (c) | <i>Principal</i> | \$ 650.50 |
| | <i>Interest Rate</i> | 5.25 % |
| | <i>Time</i> | 1 year |
| | <i>Interest Earnings</i> | _____ |
| | <i>Amount in Account</i> | _____ |
| | | |
| (d) | <i>Principal</i> | \$ 25,600.00 |
| | <i>Interest Rate</i> | 6.30 % |
| | <i>Time</i> | 10 years |
| | <i>Interest Earnings</i> | _____ |
| | <i>Amount in Account</i> | _____ |
| | | |
| (e) | <i>Principal</i> | \$ 9,286.42 |
| | <i>Interest Rate</i> | 2.75 % |
| | <i>Time</i> | 9 months |
| | <i>Interest Earnings</i> | _____ |
| | <i>Amount in Account</i> | _____ |
| | | |
| (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? | |
| | <i>Interest Earnings</i> | _____ |
| | <i>Amount in Account</i> | _____ |
-

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

<i>Employee Number</i>	25443
<i>Number of Hours Worked This Week</i>	38
<i>Hourly Wage</i>	\$ 8.75
<i>Gross Pay</i>	\$ 332.50
<i>Total Tax Deductions</i>	\$ 59.86
<i>Net Pay</i>	\$ 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 the UNIX emacs Text Editor and Type the Program Code**

Open a new UNIX session and, at the UNIX command prompt, type the following command to invoke the emacs editor.

```
emacs lab1prj4.cpp
```

When the emacs editor opens, type the program code that will prompt the user for the necessary input itmes, compute any necessary calculations and then display the output to the user. Include your name in the heading oprotion of your program code.

STEP 2 **Compile the Program**

After you have typed the given program, click **Tools** on the emacs menu and then select **Compile**. The cursor will shift to the bottom of the screen and the following will be displayed:

Student Name _____

Section _____

PROJECT FOUR

Compile command: `make -k`

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj4.cpp -o lab1prj4.cpp
```

STEP 3 Test your Program

After 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 and run your program again.

STEP 4 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 5 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 6 Close emacs and Return to the UNIX Prompt

Before proceeding to the next program, close your emacs screen to return to the UNIX prompt. You are ready to proceed to the next 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.

$$s = \frac{1}{2}gt^2$$

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 second²

Steps To Complete This Project**STEP 1** **Open the UNIX emacs Text Editor and Type the Program Code**

Open a new UNIX session and, at the UNIX command prompt, type the following command to invoke the emacs editor.

```
emacs lab1prj5.cpp
```

When the emacs editor opens, type the program code that will prompt the user for the necessary input items, compute any necessary calculations and then display the output to the user. Include your name in the heading portion of your program code.

STEP 2 **Compile the Program**

After you have typed the given program, click **Tools** on the emacs menu and then select **Compile**. The cursor will shift to the bottom of the screen and the following will be displayed:

```
Compile command: make -k
```

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj5.cpp -o lab1prj5.cpp
```

STEP 3 **Test your Program**

Before proceeding to the next program, close your emacs screen to return to the UNIX prompt. You are ready to proceed to the next project.

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

Student Name _____

Section _____

PROJECT FIVE

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	16.1
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.

STEP 6 **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 7 **Close emacs and Return to the UNIX Prompt**

Before proceeding to the next program, close your emacs screen to return to the UNIX prompt. You are ready to proceed to the next 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 the UNIX emacs Text Editor and Type the Program Code***

Open a new UNIX session and, at the UNIX command prompt, type the following command to invoke the emacs editor.

```
emacs lab1prj6.cpp
```

When the emacs editor opens, type the program code that will prompt the user for the necessary input items, compute any necessary calculations and then display the output to the user. Include your name in the heading portion of your program code.

STEP 2 *Compile the Program*

After you have typed the given program, click **Tools** on the emacs menu and then select **Compile**. The cursor will shift to the bottom of the screen and the following will be displayed:

```
Compile command: make -k
```

Erase the text `make -k` and type the following compile command in its place.

```
g++ lab1prj6.cpp -o lab1prj6.cpp
```

STEP 3 *Test your Program*

Test your program using the appropriate data.

STEP 4 *Print Your Program Code*

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

STEP 5 *Close emacs and Return to the UNIX Prompt*

Before proceeding to the next program, close your emacs screen to return to the UNIX prompt.