# Group 3: Nguyễn Đức Duy Nguyễn Công Hoàng Trần Quốc Bảo

Project X

## Checkpoint

#### 1. The following program skeleton determines whether a person qualifies for a credit card. To qualify, the person must have worked on his or her current job for at least two years and make at least $17,000 per year. Finish the program by writing the definitions of the functions qualify and noQualify . The function qualify should explain that the applicant qualifies for the card and that the annual interest rate is 12%. The function noQualify should explain that the applicant does not qualify for the card and give a general explanation why. (6.4)

#include <iostream>

**using namespace** std;

**void** qualify() {

cout << "You are qualified for a credit card.\n"

<< "Your annual interest rate is 12%.";

}

**void** noQualify() {

cout << "You are not qualified for a credit card.\n"

<< "In order to be qualified you need:\n"

<< "\t+ Annual Income at least $17,000.00\n"

<< "\t+ Continuously working on the current job for at least the last 2 years.\n";

}

**int** main() {

**double** salary;

**int** years;

cout << "This program will determine if you qualify\n";

cout << "for our credit card.\n";

cout << "What is your annual salary? ";

cin >> salary;

cout << "How many years have you worked at your ";

cout << "current job? ";

cin >> years;

**if** (salary >= 17000.0 && years >= 2)

qualify();

**else**

noQualify();

**return** 0;

}

#### 2. Write a function named timesTen The function should have an integer parameter named number . When timesTen is called, it should display the product of number times ten. (Note: just write the function. Do not write a complete program.) (6.6)

**void** timesTen(**int** number) {

cout << number\*10 << endl;

}

#### 3.The following program skeleton asks for the number of hours you’ve worked and your hourly pay rate. It then calculates and displays your wages. The function showDollars , which you are to write, formats the output of the wages. (6.10)

#include <iostream>  
#include <iomanip>

**using namespace** std;

**void** showDollars(**double**); // Function prototype

**int** main()

{

**double** payRate, hoursWorked, wages;

cout << "How many hours have you worked? ";

cin >> hoursWorked;

cout << "What is your hourly pay rate? ";

cin >> payRate;

wages = hoursWorked \* payRate;

showDollars(wages);

**return** 0;

}

// You must write the definition of the function showDollars

// here. It should take one parameter of the type double.

// The function should display the message "Your wages are $"

// followed by the value of the parameter. It should be displayed

// with 2 places of precision after the decimal point, in fixed

// notation, and the decimal point should always display.

**void** showDollars(**double** wage) {

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

cout << "Your wages are $" << wage;

}

## Programming Challenges

#### 1. Markup Write a program that asks the user to enter an item’s wholesale cost and its markup percentage. It should then display the item’s retail price. For example: - If an item’s wholesale cost is 5.00 and its markup percentage is 100%, then the item’s retail price is 10.00.

#### - If an item’s wholesale cost is 5.00 and its markup percentage is 50%, then the item’s retail price is 7.50. The program should have a function named calculateRetail that receives the wholesale cost and the markup percentage as arguments, and returns the retail price of the item. Input Validation: Do not accept negative values for either the wholesale cost of the item or the markup percentage.

#include <iostream>

#include <iomanip>

**using namespace** std;

**double** calculateRetail(**double**, **double**);

**int** main() {

**double** cost, markup100;

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

cout << "Item wholesale cost:\t";

**do** {

cin >> cost;

**if** (cost < 0)

cout << "Wholesale cost cannot be negative. Enter again:\t";

} **while** (cost < 0);

cout << "Item markup percentage (20% -> 20):\t";

**do** {

cin >> markup100;

**if** (markup100 < 0)

cout << "Markup percentage cannot be negative. Enter again:\t";

} **while** (markup100 < 0);

cout << "\n\nRetail price of the item:\t$"

<< calculateRetail(cost, markup100 / 100);

**return** 0;

}

**double** calculateRetail(**double** cost, **double** markup) {

**return** cost \* (1 + markup);

}

#### 2. Rectangle Area—Complete the Program The Student CD contains a partially written program named AreaRectangle.cpp . Your job is to complete the program. When it is complete, the program will ask the user to enter the width and length of a rectangle, and then display the rectangle’s area.

#### The program calls the following functions, which have not been written: - getLength – This function should ask the user to enter the rectangle's length, and then return that value as a double . - getWidth - This function should ask the user to enter the rectangle's width, and then return that value as a double . - getArea – This function should accept the rectangle's length and width as arguments, and return the rectangle's area. The area is calculated by multiplying the length by the width. - displayData – This function should accept the rectangle's length, width, and area as arguments, and display them in an appropriate message on the screen.

#include <iostream>  
#include <iomanip>

**using namespace** std;

**double** getLength();

**double** getWidth();

**double** getArea(**double**, **double**);

**void** displayData(**double**, **double**, **double**);

**int** main() {

**double** l = getLength(), w = getWidth();

cout << "\n\n";

displayData(l, w, getArea(l, w));

**return** 0;

}

**double** getLength() {

**double** l;

cout << "Enter rectangle length:\t";

**do** {

cin >> l;

**if** (l <= 0)

cout << "Length must > 0. Enter again:\t";

} **while** (l <= 0);

**return** l;

}

**double** getWidth() {

**double** w;

cout << "Enter rectangle width:\t";

**do** {

cin >> w;

**if** (w <= 0)

cout << "Width must > 0. Enter again:\t";

} **while** (w <= 0);

**return** w;

}

**double** getArea(**double** width, **double** height) {

**return** width \* height;

}

**void** displayData(**double** length, **double** width, **double** area) {

**const int** OFFSET = 20;

**int** m\_l = **static\_cast**<**int**> (length);

**int** m\_w = **static\_cast**<**int**> (width);

**bool** b = **false**;

m\_l = m\_l > 14 ? m\_l : 14;

m\_w = m\_w > 5 ? m\_w : 5;

cout << setw(OFFSET) << "" << setw(m\_l) << "Length: " << length << endl;

**for** (**int** i = 0; i < m\_w; i++) {

**if** (i == (m\_w - 1) / 2)

cout << setw(OFFSET - 3 - 6) << "Width: " << setw(6) << left << width << " ";

**else**

cout << setw(OFFSET) << "";

**for** (**int** j = 0; j < m\_l; j++) {

**if** (i == 0 || i == m\_w - 1 || j == 0 || j == m\_l - 1)

cout << "\* ";

**else if** (i == (m\_w - 1) / 2) {

**if** (!b)

cout << setw(m\_l - 2) << right<< " Area: "

<< setw(6) << left << area

<< setw((m\_l - 2) - 6) << "";

b = **true**;

} **else**

cout << " ";

}

cout << "\n";

}

}

#### 3.