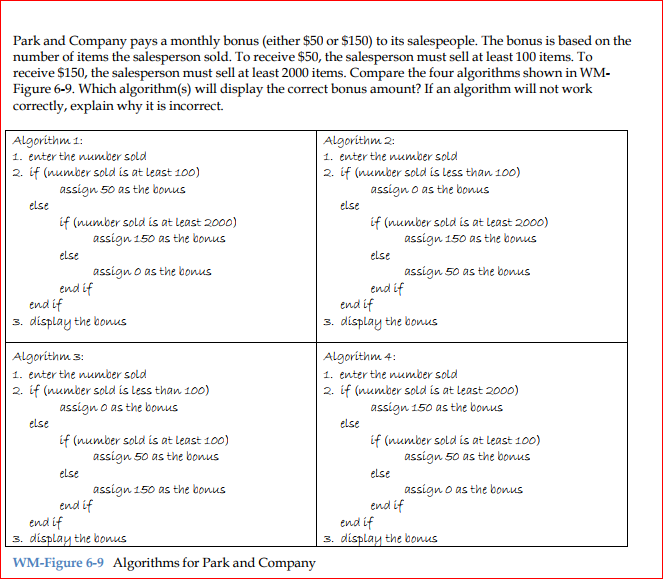
Ch 6 Exercises – Instruction Sheet

# Part 1: (10 points)



Algorithm 1: doesn’t work because it will assign a $50 bonus to everything above 100

Algorithm 2: works

Algorithm 3: eliminates everything less than 100, but doesn’t work because it will assign a $50 bonus to everything above 100

Algorithm 4: works

# Part 2: (10 points)

1. int empCode = 0;

cout << “Enter employee code: “;

cin >> empCode;

if (empCode == 1)

cout << “Sam”;

else if (empCode == 2)

cout << “Jack”;

else if (empCode == 3 || empCode == 4)

cout << “Sue”;

else if (empCode == 5)

cout << “Mary”;

else

cout << “Invalid code”;

2.

int empCode = 0;

cout << “Enter employee code: “;

cin >> empCode;

Switch (empCode)

{

case 1:

cout << “Sam”;

break;

case 2:

cout << “Jack”;

break;

case 3:

case 4:

cout << “Sue”;

case 5:

cout << “Mary”;

break;

default:

cout << “Invalide code”;

} // end switch

1. double salary = 0.0;

char empCode = “0”;

double gross = 0.0;

if (empCode == “1”)

gross = salary / 12;

else if (empCode == “2”)

gross = salary / 24;

else if (empCode == “3”)

gross = salary / 52;

else

cout << “Invalid code”;

cout << “Gross pay: “ << gross << endl:

4.

double salary = 0.0;

char empCode = “0”;

double gross = 0.0;

Switch (code)

{

case 1:

gross = salary / 12;

break;

case 2:

gross = salary / 24;

break;

case 3:

gross = salary / 52;

default:

cout << “Invalid code”;

} // end switch

cout << “Gross pay: ” << gross << endl;

# Part 3: (20 points)

Use the **Lab6-4.cpp** file inside your \Chap06\Lab6-4 Project directory for this assignment. The program in this lab should display the price of a move ticket. Tickets prices are based on the customer’s age. If the user enters a negative number, the program should display the “Invalid Age” message. to complete the following:

* Put existing C++ instructions in order and add any missing instructions.
* Test the program using 1, 3, 4, 64, 65, 70, and -3.
* Debug the program as necessary so it works as expected.
* Save the executable file as ticPrice.exe.

**Age (years) Price (in $)**

0-3 0

4-64 9

65+ 6

# Part 4: (20 points)

Complete Introductory 13 computer exercise on page 191. You do NOT have to create the folder structure as indicated nor do you have to submit an IPO chart. You need only to submit a .cpp file named **Ch6Intro13.cpp**.

# Part 5: (20 points)

Complete Intermediate 16 computer exercise on page 193. You do NOT have to create the folder structure as indicated nor do you have to submit an IPO chart. You need only to submit a .cpp file named **Ch6Inter16.cpp**.

# Part 6: (20 points)

Complete Advance 21 computer exercise on page 194. You do NOT have to create the folder structure as indicated nor do you have to submit an IPO chart. You need only to submit a .cpp file named **Ch6Adv21.cpp**.

# What to submit:

This file with responses to Part 1 and Part 2.

Lab6-4.cpp

Ch6Intro13.cpp

Ch6Inter16.cpp

Ch6Adv21.cpp

Create a directory named yourname and the chapter (example: mrodenCh6), copy the files into the directory and zip it. Upload the zip into the Ch6 Exercises assignment area.