**CSC 1100 – Problem Solving and Programming**

**Homework 3 – rory lange**

**25 points – Due February 18, 11am**

**Late deadline is February 20, 11:59pm, but 20% off**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document to the Canvas item where you downloaded this document.

**1) [2 points]** Using the February 2021 index available at web page [tiobe.com/index.php/content/paperinfo/tpci/index.html](http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html), list:

● The twelfth most popular programming language.

► groovy

● The Programming Language of the Year for 2007.

► python

**2) [2 points]** Declare the following constants:

**a)** The Kansas sales tax (use three decimal places).

► const float kansasTax = 0.065

**b)** The atomic number of iron.

► const int iron = 26

**3) [3 points]** Write formatted output statements to print the following three lines and two columns of data. Each line has a comma separating one column from the next:

State, Date founded

Michigan, 26 Jan 1837

Ohio, 1 Mar 1803

Cout << setw(25) << left << “State” << “…………………………” << setw(15) << right << “Date founded” << endl;

Cout << setw(25) << left << “Michigan” << “…………………………” << setw(15) << right << “26 Jan 1837” << endl;

Cout << setw(25) << left << “Ohio” << “…………………………” << setw(15) << right << “1 Mar 1803” << endl;

Make the columns wide enough so that no data is truncated. Left-justify the first column and right-justify the second column. Print leading dots between the two columns:

**(column-1).......(column-2)**

**4) [4 points]** What happens when a **cin statement** attempts to read the given expressions into the given variables? Use the **cin table** provided in the session notes.

1. A **float** expression is read into an **int** variable.
   1. Float is truncated
2. A **int** expression is read into a **string** variable.
   1. Converted to a string
3. A **string** expression is read into a **float** variable.
   1. 0
4. An **char** expression is read into a **string** variable.
   1. Converted to a string

**5) [2 points]** Pick a sample application from Canvas that prompts the user for a number, run the app, and when prompted enter a string. Describe what happens with the app after the string is entered.

Sample app ► Rectangle perimeter and area calculator

//==========================================================

//

// Title: Rectangle Perimeter and Area Calculator, v1

// Description:

//   This C++ console application calculates the perimeter

// and area of a rectangle.  It prompts for and gets from

// the user the length and width of the rectangle, and

// calculates and prints the area and perimeter.

//

//==========================================================

#include <cstdlib>  // For several general-purpose functions

#include <fstream>  // For file handling

#include <iomanip>  // For formatted output

#include <iostream>  // For cin, cout, and system

#include <string>  // For string data type

using namespace std;  // So "std::cout" may be abbreviated to "cout"

int main()

{

  // Declare variables

  double length;

  double width;

  double perimeter;

  double area;

  // Show application header

  cout << "Welcome to Rectangle Perimeter and Area "

    << "Calculator, v1" << endl;

  cout << "----------------------------------------"

    << "--------------" << endl << endl;

  // Prompt for and get length

  cout << "Enter the length (meters): ";

  cin >> length;

  // Prompt for and get width

  cout << "Enter the width (meters): ";

  cin >> width;

  // Calculate perimeter and area

  perimeter = 2 \* (length + width);

  area = length \* width;

  // Show inputs and outputs

  cout << "\nLength:    " << length << " meters" << endl;

  cout << "Width:     " << width << " meters" << endl;

  cout << "Perimeter: " << perimeter << " meters" << endl;

  cout << "Area:      " << area << " square meters" << endl;

  // Show application close

  cout << "\nEnd of Rectangle Perimeter and Area "

    << "Calculator, v1" << endl;

}

App behavior after string entered ► the application window closes

**6) [2 points]** When is it better to use **getline** instead of **cin** to read from the keyboard?

It is better to use getline when you are reading multiple strings because cin leaves a whitespace at the beginning of the next input

**7) [2 points]** What is needed if an app uses **cin** followed by **getline** to read from the keyboard?

The \n at the end of the cin input is read by the getline and no data is inputted for the getline statement. A way to fix this is use cin.ignore(256, \n) to ignore the \n so that the getline does not read it.

**8) [3 points]** Write the following conditions:

**a)** A simple condition that tests whether integer variable **count** is greater than twice the value of integer variable **days**.

► if (count > 2 \* days) {

}

**b)** A compound condition that tests whether char variable **code** is equal to '@' or '%'.

► if (code == “@” || code == “%”) {

}

**c)** A compound condition that tests whether string variable **lastName** is between "D" and "T".

► char lastName;

char high = 74;

char low = 64;

cout << "enter a name";

cin >> lastName;

cout << endl;

if (lastName >= low && lastName <= high) {

cout << "1";

}

**9) [2 points]** What is the primary difference between a **switch statement** and an **if statement**?

Design preference.

**10) [3 points]** Given the following **if statement**, write the equivalent logic using a ternary operator inside a **cout statement**.

if (week <= 8)

cout << "You're in the first half of the journey." << endl;

else

cout << "You're in the second half of the journey." << endl;

cout << ((week <= 8) ? "You're in the first half of the journey." : "You're in the second half of the journey.") << endl;