Ranoa, Julius
CSC 121 001 Computer Science I
September 7, 2017 Thursday

Homework on Chapter 4.
Part I – Review Questions and Exercises. Qn. 11 – 20, 22 – 24.

11. The AND logical operator works best when testing a number to determine if it is within a range.
12. The OR logical operator works best when testing a number to determine if it outside a range.
13. A variable with LOCAL or BLOCK scope is only visible when the program is executing in the block containing the variable's definitions.
14. The expression that is tested by a switch statement must have a(n) INTEGER value.
    The data types, CHAR and ENUM, also have integer values so they're valid too.
15. A program will "fall through" to the following case section if it is missing the BREAK statement.
16. What value will be stored in the variable t after each of the following statements executes?
    (A) `t = (12 > 1);` True
    (B) `t = (2 < 0);` False
    (C) `t = (5 == (3 * 2));` False
    (D) `t = (5 == 5);` True

17. Write an if statement that assigns 100 to x when y is equal to 0.

```
if (y == 0) x = 100;
```

18. Write an if/else statement that assigns 0 to x when y is equal to 10. Otherwise it should assign 1 to x.

```
if (y == 10) x = 0;
else x = 1;
```

19. Write an if/else statement that prints "Excellent" when score is 90 or higher, "Good" when score is between 80 and 89, and "Try Harder" when score is less than 80.

```
if (score >= 90) cout << "Excellent";
else if (score >= 80) cout << "Good";
else cout << "Try Harder";
// assuming score is declared int.
```

20. Write an if statement that sets the variable hours to 10 when the flag variable minimum is set to true.

```
if (minimum) hours = 10;
```

22. Convert the following if/else statements into a switch statement:

```
if (choice == 1)
    cout << fixed << showpoint << setprecision(2);
else if ((choice == 2) || (choice == 3))
    cout << fixed << showpoint << setprecision(4);
else if (choice == 4)
    cout << fixed << showpoint << setprecision(6);
else
    cout << fixed << showpoint << setprecision(8);
```

```
switch (choice) {
    case 1:
        cout << fixed << showpoint << setprecision(2);
        break;
    case 2:
    case 3:
        cout << fixed << showpoint << setprecision(4);
        break;
    case 4:
        cout << fixed << showpoint << setprecision(6);
        break;
    default:
        cout << fixed << showpoint << setprecision(8);
}
```

23. Assume the variables x = 5, y = 6, and z = 8. Indicate if each of the following conditions is true or false:

    (A) `(x == 5) || (y > 3)` True

    (B) `(7 <= x) && (z > 4)` False

    (C) `(2 != y) && (z != 4)` True

24. Assume the variables x = 5, y = 6, and z = 8. Indicate each of the following conditions is true or false.

    (A) `(x >= 0) || (x <= y)` True

    (B) `(z – y) > y` False

    (C) `!((z – y) > x)` True

Part II – Programming Challenge 11. Geometry Calculator

In submission: C++ Source Code in Three (3) Files
Screenshots of Runtime

(1) **main.cpp**

```
#include <iostream>
#include "GeometryCalculator.h"

int main() {
    GeometryCalculator me;
    me.runZeCalculator();

    return 0;
}
```

(2) **GeometryCalculator.h**

```
class GeometryCalculator {

public:
    // Constants
    enum MENU_ITEMS {
        CIRCLE = 1,
        RECTANGLE = 2,
        TRIANGLE = 3,
        EXIT = 4
    };
    const double PI = 3.14159;
    const int AREA_PRECISION = 1;

    void runZeCalculator();
    int getMenuFeedback();
    void goNuts();
    void sayBye();

    bool isMeasureValid(double input);

    void doCircleMethod();
    double getCircleArea(double radius);

    void doRectangleMethod();
    double getRectArea(double length, double width);

    void doTriangleMethod();
    double getTriangleArea(double base, double height);

};
```

(3) **GeometryCalculator.cpp**

```
#include <iostream>
#include <iomanip>
#include "GeometryCalculator.h"
using namespace std;

void GeometryCalculator::runZeCalculator() {
    switch (getMenuFeedback()) {
        case CIRCLE:
            doCircleMethod();
            sayBye();
            break;
        case RECTANGLE:
            doRectangleMethod();
            sayBye();
            break;
        case TRIANGLE:
            doTriangleMethod();
            sayBye();
            break;
```

* *GeometryCalculator.cpp continued next page*

```
        default:
            goNuts();
    }

    return;
}

void GeometryCalculator::goNuts() {
    cout << "Whoops!" << endl;
    cout << "The program doesn't know what to do now. " << endl;
    cout << "You must have entered an invalid input. " << endl << endl;

    cout << "Please close this program and try again. " << endl;
    return;
}

void GeometryCalculator::sayBye() {
    cout << "Goodbye! " << endl;
    cout << "This program has done its job. " << endl;

    return;
}

int GeometryCalculator::getMenuFeedback() {
    int choice;

    cout << "Geometry Calculator." << endl << endl;

    cout << "     1. Calculate the Area of a Circle" << endl;
    cout << "     2. Calculate the Area of a Rectangle" << endl;
    cout << "     3. Calculate the Area of a Triangle" << endl;
    cout << "     4. Quit" << endl;

    cout << endl << "Enter your choice (1-4): ";
    cin >> choice;

    cout << endl;
    return choice;
}

bool GeometryCalculator::isMeasureValid(double input) {
    if (input >= 0) return true;
    else return false;
}
```

** GeometryCalculator.cpp continued next page*

```
void GeometryCalculator::doCircleMethod() {
    double radius;

    cout << "To calculate the area of a circle, "
         << "we need the radius. " << endl;
    cout << "Please enter the value as prompted." << endl << endl;

    cout << "    Radius? ";
    cin >> radius;
    cout << endl;

    if (isMeasureValid(radius)) {
        cout << fixed << showpoint << setprecision( AREA_PRECISION );
        cout << "Yay! The area of the circle is " << getCircleArea(radius) << " square
units!" << endl << endl;
    } else goNuts();

    return;
}

double GeometryCalculator::getCircleArea(double radius) {
    return PI * radius * radius;
}

void GeometryCalculator::doRectangleMethod() {
    double length, width;

    cout << "To calculate the area of a rectangle, "
         << "we need the length and width. " << endl;
    cout << "Please enter the value as prompted." << endl << endl;

    cout << "    Length? ";
    cin >> length;
    cout << "    Width? ";
    cin >> width;
    cout << endl;

    if (isMeasureValid(length) && isMeasureValid(width)) {
        cout << fixed << showpoint << setprecision( AREA_PRECISION );
        cout << "Yay! The area of the rectangle is "
             << getRectArea(length, width)
             << " square units!" << endl << endl;
    } else goNuts();

    return;
}

double GeometryCalculator::getRectArea(double length, double width) {
    return length * width;
}
```

** GeometryCalculator.cpp continued next page*

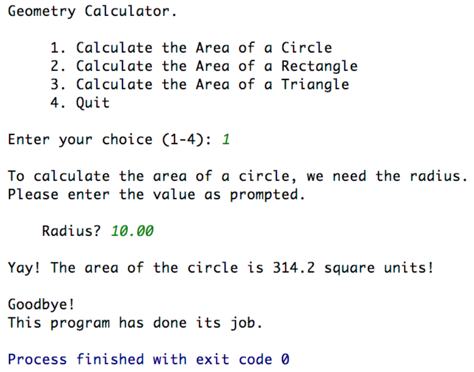
```
void GeometryCalculator::doTriangleMethod() {
    double base, height;

    cout << "To calculate the area of a triangle, "
         << "we need the base and height. " << endl;
    cout << "Please enter the value as prompted." << endl << endl;

    cout << "    Base? ";
    cin >> base;
    cout << "    Height? ";
    cin >> height;
    cout << endl;

    if (isMeasureValid(base) && isMeasureValid(height)) {
        cout << fixed << showpoint << setprecision( AREA_PRECISION );
        cout << "Yay! The area of the triangle is "
             << getTriangleArea(base, height)
             << " square units!" << endl << endl;
    } else goNuts();

    return;
}

double GeometryCalculator::getTriangleArea(double base, double height) {
    return (base * .5) * height;
}
```

Screenshots of Runtime.

Attached: 5 Images.

Image 1

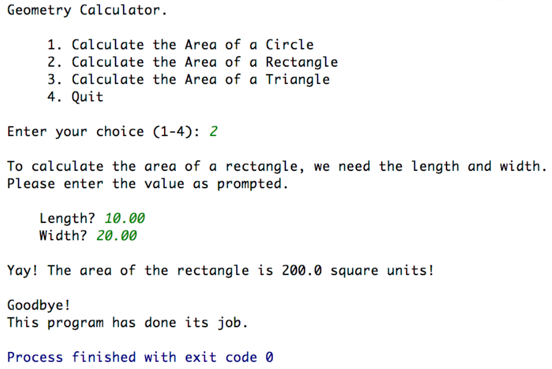
Calculating Area of a Circle



```
Geometry Calculator.

     1. Calculate the Area of a Circle
     2. Calculate the Area of a Rectangle
     3. Calculate the Area of a Triangle
     4. Quit

Enter your choice (1-4): 1

To calculate the area of a circle, we need the radius.
Please enter the value as prompted.

    Radius? 10.00

Yay! The area of the circle is 314.2 square units!

Goodbye!
This program has done its job.

Process finished with exit code 0
```

Image 2

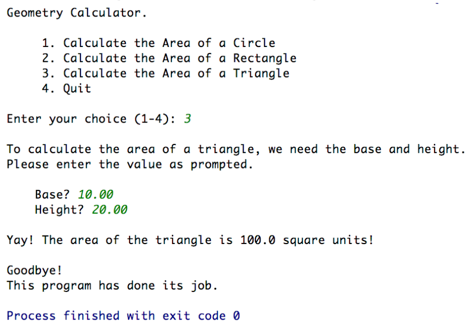
Calculating Area of a Rectangle



```
Geometry Calculator.

     1. Calculate the Area of a Circle
     2. Calculate the Area of a Rectangle
     3. Calculate the Area of a Triangle
     4. Quit

Enter your choice (1-4): 2

To calculate the area of a rectangle, we need the length and width.
Please enter the value as prompted.

    Length? 10.00
    Width? 20.00

Yay! The area of the rectangle is 200.0 square units!

Goodbye!
This program has done its job.

Process finished with exit code 0
```

Image 3

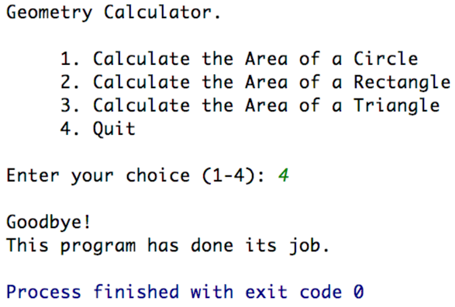
Calculating Area of a Triangle



```
Geometry Calculator.

     1. Calculate the Area of a Circle
     2. Calculate the Area of a Rectangle
     3. Calculate the Area of a Triangle
     4. Quit

Enter your choice (1-4): 3

To calculate the area of a triangle, we need the base and height.
Please enter the value as prompted.

    Base? 10.00
    Height? 20.00

Yay! The area of the triangle is 100.0 square units!

Goodbye!
This program has done its job.

Process finished with exit code 0
```

Image 4

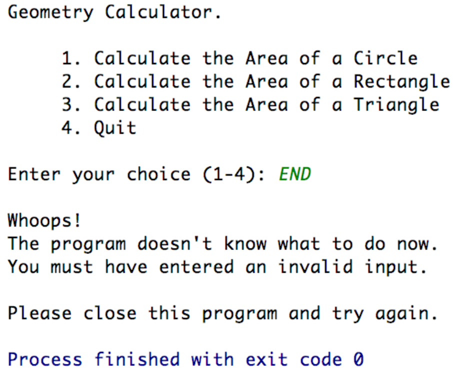
Quitting the Program



```
Geometry Calculator.

     1. Calculate the Area of a Circle
     2. Calculate the Area of a Rectangle
     3. Calculate the Area of a Triangle
     4. Quit

Enter your choice (1-4): 4

Goodbye!
This program has done its job.

Process finished with exit code 0
```

Image 5

Entering an invalid input



```
Geometry Calculator.

     1. Calculate the Area of a Circle
     2. Calculate the Area of a Rectangle
     3. Calculate the Area of a Triangle
     4. Quit

Enter your choice (1-4): END

Whoops!
The program doesn't know what to do now.
You must have entered an invalid input.

Please close this program and try again.

Process finished with exit code 0
```