National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	**Programming Fundamentals**	**Course Code:**	**CS1002**
	Program:	**Electrical Engineering**	**Semester:**	**Fall 2023**
	Assigned on:	**30 October 2023**	**Total Marks:**	**50**
	Deadline:	**7 November 2023**	**Weight:**	**3.33**
	Section:	**EE-1A and EE-1C**	**Page(s):**	**2**
	Exam Type:	**Assignment-2 Solution**	**CLO #**	**3**

Instruction:
1. Do not forget to write your Name and Roll Numbers.
2. Submit hand-written hard copy at the Start of the Class on Tuesday, 7 November.
3. **No Late submissions**. Plagiarism/copying cases to be referred to the DC.

Question No. 1 **Marks: 10**

Suppose that a solid **closed** cylinder is given to you, and the only known information is the **area of its base** and the **height**. You are required to find out this cylinder's **radius**, **circumference**, **surface area**, and **volume** using **four** interrelated functions.

The function to calculate **radius** takes the given **area** as a parameter, and returns the **radius** based on some calculations. The function to calculate **circumference** of the base, takes the **radius** as the parameter and returns the **circumference**. In a similar fashion, the function to calculate the surface area of the cylinder, takes the **circumference**, the **height**, and the **area of the base,** and returns the **surface area** of the cylinder. Lastly, the function to calculate the **volume** of the cylinder takes **radius** and **height** as parameters, and returns the **volume.**

You are required to write C++ code that implements the above functions. You may take the area and the height as inputs from the user.

For example, if a user enters area of the base as 50, and height as 10. Then you program should yield the following output.

radius = 3.99
circumference = 25.07
surface area = 350.73
volume = 500.14

Note that the above given values are approximated to 2 decimal places.

You maybe include '*cmath*' library in your program to use the built-in '*sqrt()*' function to calculate the square root of a number. Do not use any built-in C++ functions to calculate the radius, circumference, surface area, or the volume.

Solution

```
#include <iostream>
#include <cmath>

using namespace std;
```

```
// Function to calculate radius based on area
double calculateRadius(double area) {
   return sqrt(area / M_PI);
}

// Function to calculate circumference based on radius
double calculateCircumference(double radius) {
   return 2 * M_PI * radius;
}

// Function to calculate surface area of the cylinder
double calculateSurfaceArea(double circumference, double height, double baseArea) {
   return 2 * baseArea + circumference * height;
}

// Function to calculate volume of the cylinder
double calculateVolume(double radius, double height) {
   return M_PI * pow(radius, 2) * height;
}

int main() {
   // Input area and height from the user
   double baseArea, height;
   cout << "Enter the area of the base: ";
   cin >> baseArea;
   cout << "Enter the height of the cylinder: ";
   cin >> height;

   // Calculate radius
   double radius = calculateRadius(baseArea);
   // Calculate circumference
   double circumference = calculateCircumference(radius);
   // Calculate surface area
   double surfaceArea = calculateSurfaceArea(circumference, height, baseArea);
   // Calculate volume
   double volume = calculateVolume(radius, height);

   // Display the results
   cout << "Radius: " << fixed << setprecision(2) << radius << endl;
   cout << "Circumference: " << fixed << setprecision(2) << circumference << endl;
   cout << "Surface Area: " << fixed << setprecision(2) << surfaceArea << endl;
   cout << "Volume: " << fixed << setprecision(2) << volume << endl;

   return 0;
}
```

Question No. 2 **Marks: 10**

Write a program, which displays the decimal equivalent of, given binary number. Input a binary number in main function then pass this binary number to the function named **decimalConverter** which converts that number into decimal and print the decimal equivalent. Input Validation: The input must only contain 1 or 0 as its digits. In case of invalid input display on console "Conversion not possible: it is not a binary number".

Solution

```
#include <iostream>
#include <cmath>
#include <string>

using namespace std;

// Function to convert binary to decimal
int decimalConverter(const string& binaryNumber) {
   int decimal = 0;
   int binaryLength = binaryNumber.length();

   for (int i = 0; i < binaryLength; ++i) {
      // Validate input: Check if each character is '0' or '1'
      if (binaryNumber[i] != '0' && binaryNumber[i] != '1') {
         cerr << "Invalid input. Binary numbers should only contain 0s and 1s." << endl;
         exit(EXIT_FAILURE);
      }

      // Calculate decimal equivalent
      int bit = binaryNumber[i] - '0';  // Convert char to int
      decimal += bit * pow(2, binaryLength - i - 1);
   }
   return decimal;
}

int main() {
   // Input a binary number
   cout << "Enter a binary number: ";
   string binaryNumber;
   cin >> binaryNumber;

   // Call the function to convert binary to decimal
   int decimalEquivalent = decimalConverter(binaryNumber);

   // Display the result
   cout << "Decimal equivalent: " << decimalEquivalent << endl;

   return 0; }
```

Question No. 3 **Marks: 10**

Write a program that asks the user to enter an items wholesale cost and its markup percentage. It should then display the items 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.

Solution

```
#include <iostream>
using namespace std;

// Function to calculate the retail price
double calculateRetail(double wholesaleCost, double markupPercentage) {
   // Validate input
   if (wholesaleCost < 0 || markupPercentage < 0) {
      cout << "Invalid input. Please enter non-negative values for wholesale cost and markup percentage." << endl;
      return -1; // Return -1 to indicate an error
   }

   // Calculate retail price
   double retailPrice = wholesaleCost * (1 + markupPercentage / 100);
   return retailPrice;
}

int main() {
   // Input wholesale cost and markup percentage
   double wholesaleCost, markupPercentage;
   cout << "Enter the wholesale cost of the item: $";
   cin >> wholesaleCost;
   cout << "Enter the markup percentage: ";
   cin >> markupPercentage;

   // Calculate and display retail price
   double retailPrice = calculateRetail(wholesaleCost, markupPercentage);
   if (retailPrice != -1) {
      cout << "The item's retail price is: $" << retailPrice << endl;
   }

   return 0;
}
```

Question No. 4 **Marks: 10**

Write a function, which takes starting and ending integer and print the ordered pairs on the screen. Also write main function to check its functionality.

Sample output:
Enter Starting number: 1
Enter Ending number: 5
(1,1) (1,2) (1,3) (1,4) (1,5)
(2,2) (2,3) (2,4) (2,5)
(3,3) (3,4) (3,5)
(4,4) (4,5)
(5,5)

Solution

```
#include <iostream>
using namespace std;

// Function to print ordered pairs
void printOrderedPairs(int start, int end) {
    for (int i = start; i <= end; ++i) {
        for (int j = i; j <= end; ++j) {
            cout << "(" << i << "," << j << ") ";
        }
        cout << endl;
    }
}

int main() {
    // Input starting and ending numbers
    int start, end;
    cout << "Enter starting number: ";
    cin >> start;
    cout << "Enter ending number: ";
    cin >> end;

    // Check for valid input
    if (start > end) {
        cout << "Invalid input. Starting number should be less than or equal to the ending number." << endl;
        return 1; // Return 1 to indicate an error
    }

    // Call the function to print ordered pairs
    printOrderedPairs(start, end);

    return 0;
}
```

Question No. 5 **Marks: 10**

Jason opened a coffee shop at the beach and sells coffee in three sizes: small (9 oz), medium (12 oz), and large (15 oz). The cost of one small cup is $1.75, one medium cup is $1.90, and one large cup is $2.00. Write a menu-driven program that will make the coffee shop operational.

Your program should allow the user to do the following:
a. Buy coffee in any size and in any number of cups.
b. At any time show the total number of cups of each size sold.
c. At any time show the total amount of coffee sold.
d. At any time show the total money made.
Your program should consist of at least the following functions: a function to show the user how to use the program, a function to sell coffee, a function to show the number of cups of each size sold, a function to show the total amount of coffee sold, and a function to show the total money made. Your program should not use any global variables and special values such as coffee cup sizes and cost of a coffee cup must be declared as named constants.

Solution

```
#include <iostream>
#include <iomanip>
using namespace std;

// Named constants
const int SMALL_SIZE = 9;
const int MEDIUM_SIZE = 12;
const int LARGE_SIZE = 15;

const double SMALL_COST = 1.75;
const double MEDIUM_COST = 1.90;
const double LARGE_COST = 2.00;

// Function to display menu
void displayMenu() {
   cout << "Coffee Shop Menu:\n";
   cout << "a. Buy coffee\n";
   cout << "b. Show total number of cups of each size sold\n";
   cout << "c. Show total amount of coffee sold\n";
   cout << "d. Show total money made\n";
   cout << "e. Exit\n";
}

// Function to sell coffee
void sellCoffee(int& smallCups, int& mediumCups, int& largeCups, double& totalAmount, double&
totalMoney) {
   char size;
   int quantity;
```

```
    cout << "Enter the size of the coffee (s for small, m for medium, l for large): ";
    cin >> size;

    switch (size) {
        case 's':
            cout << "Enter the quantity for small size: ";
            cin >> quantity;
            smallCups += quantity;
            totalAmount += SMALL_SIZE * quantity;
            totalMoney += SMALL_COST * quantity;
            break;
        case 'm':
            cout << "Enter the quantity for medium size: ";
            cin >> quantity;
            mediumCups += quantity;
            totalAmount += MEDIUM_SIZE * quantity;
            totalMoney += MEDIUM_COST * quantity;
            break;
        case 'l':
            cout << "Enter the quantity for large size: ";
            cin >> quantity;
            largeCups += quantity;
            totalAmount += LARGE_SIZE * quantity;
            totalMoney += LARGE_COST * quantity;
            break;
        default:
            cout << "Invalid size. Please enter 's', 'm', or 'l'.\n";
    }
}

// Function to show the number of cups of each size sold
void showTotalCups(int smallCups, int mediumCups, int largeCups) {
    cout << "Total number of cups sold:\n";
    cout << "Small: " << smallCups << " cups\n";
    cout << "Medium: " << mediumCups << " cups\n";
    cout << "Large: " << largeCups << " cups\n";
}

// Function to show the total amount of coffee sold
void showTotalAmount(double totalAmount) {
    cout << "Total amount of coffee sold: " << totalAmount << " oz\n";
}
```

```
// Function to show the total money made
void showTotalMoney(double totalMoney) {
    cout << "Total money made: $" << fixed << setprecision(2) << totalMoney << endl;
}

int main() {
    int smallCups = 0, mediumCups = 0, largeCups = 0;
    double totalAmount = 0, totalMoney = 0;
    char choice;

    do {
        displayMenu();
        cout << "Enter your choice (a-e): ";
        cin >> choice;

        switch (choice) {
            case 'a':
                sellCoffee(smallCups, mediumCups, largeCups, totalAmount, totalMoney);
                break;
            case 'b':
                showTotalCups(smallCups, mediumCups, largeCups);
                break;
            case 'c':
                showTotalAmount(totalAmount);
                break;
            case 'd':
                showTotalMoney(totalMoney);
                break;
            case 'e':
                cout << "Exiting the program.\n";
                break;
            default:
                cout << "Invalid choice. Please enter a valid option (a-e).\n";
        }
    } while (choice != 'e');

    return 0;
}
```