ECE 111 – SP 23: Introduction to C and C++ Programming
Instructor: Dr. Shayan (Sean) Taheri
Gannon University (GU)

Assignments 8, 9, and 10

Total Points: 100**; and Deadline:** April/28/2023, 11:59 PM.

Note – Cheating and Plagiarism: Cheating and plagiarism are not permitted in any form and cause certain penalties. The instructor reserves the right to fail culprits.

Deliverable: All your responses to the assignment questions should be included in a single compressed file to be uploaded to the Gannon University (GU) – Blackboard Learn environment.

Important: Read the following parts from the **Chapters 8, 9, and 10** of your textbook, which is available in the GU – Blackboard Learn environment, **before** working on your assignments.

- **Chapter 8**: Pages **522-551**, **561-573**, and **575-589**.
- **Chapter 9**: Pages **612-620**.
- **Chapter 10**: Pages **652-685**.

Assignment 8: Arrays and Strings

Question 1 (12.5 pts.). Apply cryptographic technique/algorithms to problem solving: The "**Exclusive-OR (XOR) Cipher/Cryptographic Algorithm**" is an effective and easy to implement method of symmetric encryption-decryption that can protect our data from adversaries. Refer to the following links for getting information about this algorithm. Write a program in C++ programming language using the following segment of code to perform XOR encryption and decryption, and explain your answer briefly.

Links:

- XOR cipher - Wikipedia
- Encryption - Wikipedia
- Plaintext - Wikipedia
- Ciphertext - Wikipedia
- Cryptography - Wikipedia
- Symmetric-key algorithm - Wikipedia
- Public-key cryptography - Wikipedia

Code Segment:

```
// Task 1: Determine the types of the input and the output data for
Encryption-Decryption function: "INPUT_TYPE" and "OUTPUT_TYPE".

OUTPUT_TYPE EncryptDecrypt(INPUT_TYPE toEncDec) {

	// Task 2: Declare a "Char" array for the Encryption Key.

	// Task 3: Perform "Exclusive-OR (XOR) Encryption" between the input
text/data and the key.
```

```
}

int main(int argc, const char * argv[])
{
    // Task 4: Determine the type of the encrypted data (a.k.a. Ciphertext).

     ENCRYPTION_TYPE encrypted = EncryptDecrypt("Your Input Text");
    cout << "Encrypted: " << encrypted << "\n";

     // Task 5: Determine the type of the decrypted data (a.k.a. Plaintext).

    DECRYPTION_TYPE decrypted = EncryptDecrypt(encrypted);
    cout << "Decrypted: " << decrypted << "\n";

    return 0;
}
```

Question 2 (12.5 pts.). Specify the final content of "`beta`" in the following segment of code, and explain your answer briefly.

```
int beta[3][3];
for (int i = 0; i < 3; i++)
     for (int j = 0; j < 3; j++)
          beta[i][j] = i * j;
```

Question 3 (12.5 pts.). Determine the output of the following code, and explain your answer briefly.

```
const double PI = 3.14159;
double cylinderRadii[5] = {3.5, 7.2, 10.5, 9.8, 6.5};
double cylinderHeights[5] = {10.7, 6.5, 12.0, 10.5, 8.0};
double cylinderVolumes[5];
cout << fixed << showpoint << setprecision(2);
for (int i = 0; i < 5; i++)
 cylinderVolumes[i] = 2 * PI * cylinderRadii[i]
 * cylinderHeights[i];
for (int i = 0; i < 5; i++)
 cout << (i + 1) << " " << cylinderRadii[i] << " "
 << cylinderHeights[i] << " " << cylinderVolumes[i]
 << endl;
```

Question 4 (12.5 pts.). Analyze data and interpret results to validate which requirements are met: Write a program in C++ programming language that sorts the following list in both "**Ascending**" and "**Descending**" formats using the "**Selection Sort**" algorithm (i.e., discussed in the **Chapter 8**). Analyze the outputs of your code, specify whether the problem requirements were met, and explain your answer briefly.

12, 50, 68, 30, 46, 5, 92, 10, 38

Assignment 9: Records (structs)

Question 5 (12.5 pts.). Complete the following items. Explain your answer briefly.

A. Define a struct `fruitType` to store the following data about a fruit: Fruit name (`string`), color (`string`), fat (`int`), sugar (`int`), and carbohydrate (`int`).
B. Write a function, `getFruitInput` to read and store data into a variable of `fruitType`.
C. Write a function, `printFruitInfo` to output data stored into a variable of `fruitType`. Use appropriate labels to identify each component.

Question 6 (12.5 pts.). Define a struct `computerType` to store the following data about a computer: Manufacturer (`string`), model type (`string`), processor type (`string`), random-access memory (RAM) size (`int`) in gigabyte (GB), hard drive size (`int`) in GB, year when the computer was built (`int`), and the price (`double`). Explain your answer briefly.

Assignment 10: Classes and Data Abstraction

Question 7 (12.5 pts.). Consider the following statements and question items. Explain your answer briefly.

```
class temporary
{
public:
 void set(string, double, double);
 void print();
 double manipulate();
 void get(string&, double&, double&);
 void setDescription(string);
 void setFirst(double);
 void setSecond(double);
 string getDescription() const;
 double getFirst() const;
 double getSecond() const;
 temporary(string = "", double = 0.0, double = 0.0);
private:
 string description;
 double first;
 double second;
};
```

A. How many members does class temporary have?
B. How many private members does class temporary have?
C. How many constructors does class temporary have? Can this constructor be used to initialize an object without specifying any parameters? If yes, then illustrate with an example; otherwise explain why it cannot be used to initialize an object without specifying any parameters.

Question 8 (12.5 pts.). Consider the following declarations and question items. Explain your answer briefly.

```
class houseType
{
public:
 void set(string, int, int, int, int, int, double, double);
 void print() const;
 void setStyle(string);
 string getStyle() const;
 void setNumOfBedrooms(int);
 int getNumOfBedrooms() const;
 void setNumOfBathrooms(int);
 int getNumOfBathrooms() const;
 void setNumOfCarsGarage(int);
 int getNumOfCarsGarage() const;
 void setYearBuilt(int);
 int getYearBuilt() const;
 void setFinishedSquareFootage(int);
 int getFinishedSquareFootage() const;
 void setPrice(double);
 double getPrice() const;
 void setTax(double);
 double getTax() const;
 houseType(string = "", int = 0, int = 0, int = 0, int = 0,
 int = 0, double = 0, double = 0);
private:
 string style;
 int numOfBedrooms;
 int numOfBathrooms;
 int numOfCarsGarage;
 int yearBuilt;
 int finishedSquareFootage;
 double price;
 double tax;
};

houseType newHouse; //variable declaration
```

A. How many members does class `houseType` have?
B. How many private members does class `houseType` have?
C. How many constructors does class `houseType` have?
D. How many constant functions does class `houseType` have?