Structured Programming Course Project

Instructions:

1. Choose one of the following three projects as your course project. Or you could choose any other topic you like with my approval.

2. You are allowed to form a team. But a team cannot have more than three people. The more team members, the higher expectation I would have.

3. Put your team member names in the shared google doc. (I will send you the link)

4. What you need to deliver:

- A .C code file. Properties of a good code: (a) works; (b) written in a good style with proper and consistent indent; (c) have necessary comments to help you and other programmers to navigate through the code;

- A documentation file (one or two pages). It should clearly state any applicable steps of the software development method that you take to develop the program. When you write the documentation file, position yourself as a project manager. You want to give necessary information for a software engineer to implement the code and for yourself to develop version 2.0 in the future. A flow structure may be helpful.

- A presentation. Your team will present the project in a briefly and clearly manner on the last day of class (presentation time = 90 minutes / number of teams). You could pretend you are at a new product release conference. Try to sell the product to conference attendees. After all, every effort we are making is for business.

Project mark scheme: (15 marks)

10 Marks – Content of Code:

- Did the code achieve what it set out to do? Is it written in a good style? Is it clearly commented? Were the algorithms used suitable? Does it have a user-friendly interface? The more complex the task your code achieves the better your mark in this section.

3 Marks – Documentation:

- Did you clearly state how the program is developed? Do you have any thoughts on new features that you would like to add to version 2.0?

2 Marks – Presentation:

- How to use the program? What is the importance of the program? Be professional.

Project I: Enforcing Privacy with Cryptography

The explosive growth of Internet communications and data storage on Internet-connected computers has greatly increased privacy concerns. The field of cryptography is concerned with coding data to make it difficult (and hopefully-with the most advanced schemes-impossible) for unauthorized users to read. In this project, you'll investigate a simple scheme for encrypting and decrypting data. A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely. All the data is transmitted as four-digit integers. Your application should read a four-digit integer entered by the user and encrypt it as follows:

1. Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10.

2. Swap the first digit with the third, and swap the second digit with the fourth.

3. Then print the encrypted integer.

Write a separate application that inputs an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number.

Note: You are encouraged to research "public key cryptography" in general and the PGP (Pretty Good Privacy) specific public key scheme. You may also want to investigate the RSA scheme, which is widely used in industrial-strength applications. It would be the best if you could implement a practical cryptography scheme.

Project II: Guess the Number

Write a C program that plays the game of "guess the number" as follows: Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 20. The program then types:
I have a number between 1 and 20.
Can you guess my number?
Please type your first guess.

The player then types a first guess. The program responds with one of the following:
1. Excellent! You guessed the number! Would you like to play again (1 or 0) ?
2. Too low. Try again.
3. Too high. Try again.

If the player's guess is incorrect, your program should loop until the player finally gets the number right. Your program should keep telling the player "Too high" or "Too low" to help the player find the correct answer. Count the number of guesses the player makes. If the number is less than 5, print "You are awesome!" If the player guesses the number in 5 tries, then print "You are a good guesser!" If the player makes more than 5 guesses, then print "You should be able to do better!"

Note: The searching technique employed in this program is called binary search. You may want to do some research on it and present this technique as a play tip in your project presentation. You could also add more features to the game. For example, at the beginning of the game, ask a user to choose the difficulty level of the game. If the difficulty level is higher, the range of the number to be guessed is larger.

Project III: Computer Assisted Instruction

The use of computers in education is referred to as computer-assisted instruction. Write a program that helps an elementary school student learn addition, subtraction, multiplication and division. At the beginning, the main menu allows a user to choose which operator to learn. Then a random question pops up such as

How much is 6 times 7?

The student then types the answer. Your program checks the student's answer. If it is correct, print "very good!", then ask the user whether repeat the same operator or go back to the main menu. If the answer is wrong, print "No, please try again.", then let the student try the same question repeatedly until the student finally gets it right.

Other topics:

- Blackjack (twenty-one)
- Roulette
- Three card poker
- Mortgage calculator
- Investing calculator
- Or choose your own