**Jaypee University of Engineering and Technology, Guna**

**Department of Computer Science and Engineering**

**Object Oriented Programming (18B11CI211)**

**Assignment-1**

**Floating Date: 15/03/2021 Submission Date: 28/03/2021**

**Note: No assignment will be accepted after deadline. Zero marks will be awarded if any plagiarism is found.**

Q1. Explain following concepts by writing C++ programs for each:

1. Friend function
2. Friend class
3. Static data member and static member function
4. Constant member function
5. Default parameters
6. Array of objects
7. Reference variable
8. Passing object to a function

Q2. Suppose you have to simulate “coin tossing”. For each toss of the coin, the program should print Heads or Tails. Let the program toss the coin 100 times and count the number of times each side of the coin appears. Print the results. The program should call a separate function flip that takes no arguments and returns 0 for tails and 1 for heads. [Note: If the program realistically simulates the coin tossing, then each side of the coin should appear approximately half the time.]

Q3. Suppose you have to develop small software to assist the primary students. Write a program that will help an elementary school student learn multiplication. Use the rand function to produce two positive one-digit integers. The program should then prompt the user with a question, such as

How much is 6 times 7?

The student then inputs the answer. Next, the program checks the student’s answer. If it’s correct, display the message "Very good!" and ask another multiplication question. If the answer is wrong, display the message "No. Please try again." and let the student try the same question repeatedly until the student finally gets it right. A separate function should be used to generate each new question. This function should be called once when the application begins execution and each time the user answers the question correctly.

For making your software more interesting, program should display various comments for each answer as follows:

Possible responses to a correct answer:

**Very good!**

**Excellent!**

**Nice work!**

**Keep up the good work!**

Possible responses to an incorrect answer:

**No. Please try again.**

**Wrong. Try once more.**

**Don't give up!**

**No. Keep trying.**

Use random-number generation to choose a number from 1 to 4 that will be used to select one of the four appropriate responses to each correct or incorrect answer. Use a **switch** statement to issue the responses.

To make you software more sophisticate, system should monitor the student’s performance over a period of time. The decision to begin a new topic is often based on the student’s success with previous topics. System should count the number of correct and incorrect responses typed by the student. After the student types 10 answers, your program should calculate the percentage that is correct. If the percentage is lower than 75%, display "Please ask your teacher for extra help.", then reset the program so another student can try it. If the percentage is 75% or higher, display "Congratulations, you are ready to go to the next level!", then reset the program so another student can try it. System should support various difficulty levels. At a difficulty level of 1, the program should use only single-digit numbers in the problems; at a difficulty level of 2, numbers as large as two digits, and so on.