PRACTICE QUESTIONS

1. Display a Welcome Message

Task: Write a program that displays a welcome message introducing the course and explaining the importance of learning C++.

Code:

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


int main(){
    cout << "Welcome to my program!"<<endl;
    cout << "-----------------"<<endl;
    cout << "Importance of C++"<<endl;
    cout << "C++ is a great language. It was introduced by \'Bjarne Stroustrup\'. It is an evolved version of C programmimg language, C was the first programming language."<<endl;
}
```

2. Input & Output: User Information Form

Task: Write a program that asks the user to enter their name, age, and favorite programming language, then prints this information in a formatted way.

Code:

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


int main(){
    string name, proglang, gender;
    int age = 0;
    cout << "Enter your name: ";
    getline(cin, name);
    cout << "Enter your gender (male/female/other): ";
    getline(cin, gender);
    cout << "Enter your age: ";
    cin >> age;
    cout << "Enter your favourite prigramming language: ";
    cin.ignore();
    getline(cin, proglang);

    if(gender == "male"){
```

```
        gender = "He";
    }
    else if(gender == "female"){
        gender = "She";
    }
    else{
        gender = "He/She";
    }
    cout << "***************"<<endl;
    cout << "There is a person named \'"<<name<<"\'. "<<gender<<" is "<<age<<"
years old. "<<gender<<" loves "<<proglang<<"."<<endl;



}
```

3. Conditional Statements (If-Else): Grade Calculator

Task: Develop a program that asks the user for a percentage score and determines the corresponding grade (A, B, C, D, F) using an if-else statement.

Code:

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


int main(){
    float per = 0.00;
    cout << "Enter your percentage: ";
    cin >> per;
    if(per >= 85){
        cout << "Grade A"<<endl;
    }
    else if(per < 85 && per >= 70){
        cout << "Grade B"<<endl;
    }
    else if(per < 70 && per >= 60){
        cout << "Grade C"<<endl;
    }
    else if(per < 60 && per >= 50){
        cout << "Grade D"<<endl;
    }
    else{
        cout << "Grade F"<<endl;
    }
}
```

4. Conditional Statements (If-Else): Even or Odd Checker

Task: Write a program that asks the user to enter an integer and determines whether the number is even or odd.

Code:

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


int main(){
    int num = 0;
    cout << "Enter any number: ";
    cin >> num;
    if(num%2 == 0){
        cout <<num<< " is an Even number."<<endl;
    }
    else{
        cout <<num<< " is an Odd number."<<endl;
    }
}
```

5. Decision Control using Switch Case: Basic Calculator

Task: Develop a program that takes two numbers and an operator (+, -, *, /) as input and performs the corresponding operation using a switch statement.

Code:

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


int main(){
    float n1 = 0.00, n2 = 0.00;
    char op;
    cout << "Enter first number: ";
    cin >> n1;
    cout << "Enter second number: ";
    cin >> n2;
    cout << "Enter operator(+, -, *, /): ";
    cin >> op;

    switch(op){
        case '+':
        cout << "Addition is: "<<(n1+n2)<<endl;
        break;
```

```
        case '-':
        cout << "Subtraction is: "<<(n1-n2)<<endl;
        break;
        case '*':
        cout << "Multiplication is: "<<(n1*n2)<<endl;
        break;
        case '/':
        if(n2 == 0){
            cout << "Error: Division by zero is not allowed."<<endl;
        }
        else{
            cout <<n1<<"/"<<n2<<" = "<<n1/n2<<endl;
        }
        break;
        default:
        cout << "Invalid operator!"<<endl;
    }
}
```

6. Loop Control Instructions (For Loop): Multiplication Table

Task: Write a program that takes an integer from the user and prints its multiplication table using a for loop.

Code:

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


int main(){
    int num = 0, lim = 0;
    cout << "Enter number of which you want to see table: ";
    cin >> num;
    cout << "Enter number till which you want to see multiplication: ";
    cin >> lim;

    for(int i = 1; i <= lim; i++){
        cout <<num<<" * "<<i<<" = "<<num*i<<endl;
    }
}
```

7. Loop Control Instructions (While Loop): Sum of Natural Numbers

Task: Develop a program that takes a positive integer `n` from the user and calculates the sum of all numbers from 1 to `n` using a `while` loop.

Code:

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


int main(){
    int num = 0, sum = 0;
    cout << "Enter number till which you want to see sum: ";
    cin >> num;

    int i = 1;
    while(i <= num){
        sum += i;
        i++;
    }
    cout << "Sum is: "<<sum<<endl;
}
```

8. Loop Control Instructions (Do-While Loop): Number Guessing Game

Task: Write a simple number guessing game where the user must guess a randomly generated number between 1 and 100. Use a `do-while` loop to keep prompting the user until they guess correctly.

Code:

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


int main(){
    int rand = 65, userInput = 0, turns = 0;
    do{
        cout << "System has generated a random no. b/w 1 & 100. Guess it! ";
        cin >> userInput;
        turns++;
    }while(userInput != rand);
    cout << "You gussed it in \'"<<turns<<"\' turn(s)."<<endl;
}
```

9. Nested Loops: Right-Angled Triangle Pattern

Task: Write a program that prints a right-angled triangle of stars (`*`) for a given height using nested loops. Example for `n = 4`:

```
*
**
***
****
```

Code:

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


int main(){
    int rows = 0;
    cout << "Enter no. of rows: ";
    cin >> rows;
    for(int i = 1; i <= rows; i++){
        for(int j = 1; j <= i; j++){
            cout << "* ";
        }
        cout << endl;
    }
}
```

10. Nested Loops: Number Pyramid Pattern

Task: Create a program that prints a number pyramid based on the user input. Example for `n = 4`:

```
   1
  121
 12321
1234321
```

Code:

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


int main(){
    int rows = 0;
    cout << "Enter no. of rows: ";
    cin >> rows;
    for(int i = 1; i <= rows; i++){
        for(int j = rows; j >= i; j--){
            cout << " ";
        }
        for(int k = 1, p = 1; k < i; k++, p++){
            cout <<p;
        }
        for(int l = 1, p = i; l <= i; l++, p--){
            cout <<p;
        }
        cout << endl;
    }
}
```

11. Nested Loops: Prime Number Checker

Task: Develop a program that takes an integer from the user and determines whether it is a prime number using a `for` loop with a nested condition.

Code:

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


int main(){
    int num = 0, count = 0;
    cout << "Enter any number: ";
    cin >> num;
    for(int i = 1; i <= num; i++){
        if(num%i == 0){
            count++;
        }
    }
    if(count == 2){
        cout <<num<< " is a prime number."<<endl;
    }
    else{
        cout <<num<< " is not a prime number."<<endl;
    }
}
```

12. Functions (Basic): Simple Addition Function

Task: Write a function named `addNumbers` that takes two integers as arguments and returns their sum. Call this function from `main()` and display the result.

Code:

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

float addNumbers(float a, float b){
    return (a+b);
}

int main(){
    float num1 = 0.00, num2 = 0.00, res = 0.00;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;
    res = addNumbers(num1, num2);
    cout << "Addition is: "<<res;
}
```

13. Functions: Factorial Calculator

Task: Create a function `factorial(int n)` that calculates the factorial of a number using a loop. Call this function from `main()` and display the result.

Code:

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

int factorial(int a){
    int fact = 1;
    for(int i = 1; i <= a; i++){
        fact *= i;
    }
    return fact;
}

int main(){
    int num = 0;
    cout << "Enter number of which you want to see factorial: ";
    cin >> num;
    int res = factorial(num);
    cout << "Factorial of "<<num<<" is: "<<res<<endl;

}
```

14. Functions with Multiple Parameters: Greatest of Three Numbers

Task: Write a function `findMax(int a, int b, int c)` that determines the largest of three numbers. Call this function in `main()` to display the result.

Code:

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

void findMax(int a, int b, int c){
    if(a > b && a > c){
        cout << "Largest number is: "<<a<<endl;
    }
    else if(b > a && b > c){
        cout << "Largest number is: "<<b<<endl;
    }
    else{
        cout << "Largest number is: "<<c<<endl;
    }
}

int main(){
    int a = 0, b = 0, c = 0;
    cout << "Enter three numbers: ";
    cin >> a >> b >> c;
    findMax(a, b, c);
}
```

15. Functions with Scope Demonstration: Local and Global Variables

Task: Develop a program that demonstrates the difference between local and global variables by modifying a global variable inside a function.

Code:

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

int global = 10;

void globalVar(){
    global += 10;

}

void localVar(int a){
    cout << "Actual value which was passed: "<<a<<endl;
    a += 10;
    cout << "After doing changes in user defined function: "<<a<<endl;
}


int main(){
    cout << "Actual value of global variable: "<<global<<endl;
    globalVar();
    cout << "After doing some chages in user defined function:
"<<global<<endl;

    //it's actual value was 10 but as we have changed it in function and it
was a global vriable,
    //so we got updated value

    int local = 50;
    cout << "Local variable's actual value: "<<local<<endl;
    localVar(local);
    cout << "It is still same in main function: "<<local;


}
```

16. Menu-Driven Program: Temperature Converter

Task: Write a menu-driven program that allows the user to:

1. Convert Celsius to Fahrenheit.
2. Convert Fahrenheit to Celsius.
3. Exit the program.

Use a loop to keep the menu active until the user chooses to exit.

Code:

```
#include<iostream>
using namespace std;
void celToFah(float temp){
    cout << "Temperature in Fahrenhite: "<<(temp*(9/5)+32)<<endl;
}
void fahToCel(float temp){
    cout << "Temperture in Celcius is: "<<((temp-32)*5/9)<<endl;
}
int main(){
    float temp= 0.00;
    int choice = 0;
    cout << "***Welcome to Temperature Converter***"<<endl;
    cout << endl;
    do{
        cout << "TO convert from Celsius to Fahrenhite, Press '1'"<<endl;
        cout << "TO convert from Fahrenhite to Celsius, Press '2'"<<endl;
        cout << "To exit, Press '0'"<<endl;
        cout << "Enter choice: ";
        cin >> choice;
        if(choice == 1){
            cout << "Enter temperature in celsius: ";
            cin >> temp;
            celToFah(temp);
        }
        else if(choice == 2){
            cout << "Enter temperature in fahrenhite: ";
            cin >> temp;
            fahToCel(temp);
        }
        else if(choice == 0){
            cout << "program terminated..."<<endl;
        }
        else{
            cout << "invalid choice!"<<endl;
        }
    }while(choice != 0);
}
```

17. Menu-Driven Program: Student Report System

Task: Develop a program that provides the following options:

1. Input student marks for three subjects.
2. Calculate and display the percentage.
3. Determine the grade based on the percentage.
4. Exit.

Use functions to implement each menu option.

Code:

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

float totalMarks(){
    float sub1 = 0.00, sub2 = 0.00, sub3 = 0.00;
    cout << "Enter marks of your 1st subject: ";
    cin >> sub1;
    cout << "Enter marks of your 2nd subject: ";
    cin >> sub2;
    cout << "Enter marks of your 3rd subject: ";
    cin >> sub3;
    cout << "Total obtained marks are: "<<(sub1 + sub2 + sub3)<<endl;
    return (sub1 + sub2 + sub3);
}

float percentage(){
    float obtMarks = totalMarks();
    cout << "Your percentage is: "<<((obtMarks/300)*100.00)<<endl;
    return ((obtMarks/300)*100.00);
}

void grade(){
    float per = percentage();
    if(per >= 85){
        cout << "Grade A"<<endl;
    }
    else if(per < 85 && per >= 70){
        cout << "Grade B"<<endl;
    }
    else if(per < 70 && per >= 60){
        cout << "Grade C"<<endl;
    }
    else if(per < 60 && per >= 50){
        cout << "Grade D"<<endl;
    }
    else{
```

```
        cout << "Grade F"<<endl;
    }
}

int main(){
    cout << "***Student Report System***"<<endl;
    cout << endl;
    int choice = 0;
    do{
        cout << "---MENU---"<<endl;
        cout << "To calculate total marks, Press '1'"<<endl;
        cout << "To calculate percentage, Press '2'"<<endl;
        cout << "To see grade, Press '3'"<<endl;
        cout << "To exit, Press '0'"<<endl;
        cout << "Enter choice: ";
        cin >> choice;
        if(choice == 1){
            totalMarks();
        }
        else if(choice == 2){
            percentage();
        }
        else if(choice == 3){
            grade();
        }
        else if(choice == 0){
            cout << "terminating program..."<<endl;
        }
        else{
            cout << "Invalid choice"<<endl;
        }

    }while(choice != 0);
}
```

18. Menu-Driven Program: Bank Transaction System

Task: Create a program that allows the user to:

1. Deposit money.
2. Withdraw money (with a balance check).
3. Check balance.
4. Exit.

Use loops and functions for implementation.

Code:

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

float totalBal = 0.00;

void deposit(){
    float amount = 0.00;
    cout << "Enter amount you want to deposit: ";
    cin >> amount;
    totalBal +=amount;
}

void withdraw(){
    float amount = 0.00;
    cout << "Enter amount you want to withdraw: ";
    cin >> amount;
    if(amount > totalBal){
        cout << "Insufficient balance."<<endl;
    }
    else{
        totalBal -= amount;
    }
}
void checkBal(){
    cout << "Your total balance is: "<<totalBal<<endl;
}


int main(){
    cout << "***Bank Transaction System***"<<endl;
    cout << endl;
    int choice = 0;
    do{
        cout << "---MENU---"<< endl;
        cout << "To deposit money, Press '1'"<< endl;
        cout << "To withdraw money, Press '2'"<< endl;
```

```
        cout << "To check balance, Press '3'"<< endl;
        cout << "To exit, Press '0'"<< endl;
        cout << "Enter choice: ";
        cin >> choice;
        cout << endl;
        if(choice == 1){
            deposit();
        }
        else if(choice == 2){
            withdraw();
        }
        else if(choice == 3){
            checkBal();
        }
        else if(choice == 0){
            cout << "Terminating program..."<< endl;
        }
        else{
            cout << "Invalid choice."<< endl;
        }

    }while(choice != 0);

}
```

19. Simple Billing System

Task: Develop a C++ program that simulates a basic billing system. The menu should include:

1. Add an item with its price.
2. Display the total bill.
3. Apply a discount if the total bill exceeds a certain amount.
4. Exit.

Use functions, loops, and conditional statements effectively.

Code:

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

float totalPrice = 0.00;

void addItem(){
    int item = 0;
    float price = 0.00;
    cout << "Enter quantity of items: ";
    cin >> item;
    cout << "Enter price of this item: ";
    cin >> price;
    totalPrice +=(item*price);
}

void calculateBill(){
    if(totalPrice >= 500){
        cout << "You are getting a discount of 10%"<< endl;
        totalPrice = (totalPrice - (totalPrice*0.1));
        cout << "Your new bill is: "<< totalPrice << endl;
    }
    else{
        cout << "Your total bill is: "<< totalPrice << endl;
    }
}

int main(){
    cout << "***Welcome To Super Market***"<< endl;
    cout << endl;
    int choice = 0;
    do{
        cout << "---MENU---"<< endl;
        cout << "To add an item with its price, Press '1'"<< endl;
        cout << "To calculate bill, Press '2'"<< endl;
        cout << "To exit, Press '0'"<< endl;
```

```
        cout << "Enter choice: ";
        cin >> choice;
        if(choice == 1){
            addItem();
        }
        else if(choice == 2){
            calculateBill();
        }
        else if(choice == 0){
            cout << "terminating program..."<< endl;
        }
        else{
            cout << "Invalid choice."<< endl;
        }

    }while(choice != 0);
}
```