

Module 3 - Introduction to OOPS Programming (Theory Exercise)

1 Introduction to C++

1.1 First C++ Program: Hello World

- Write a simple C++ program to display "Hello, World!".
- Objective: Understand the basic structure of a C++ program, including `#include`, `main()`, and `cout`.

```
1 #include<iostream>
2 using namespace std;
3
4 main(){
5     cout<<"Hello, World!";
6 }
```

1.2 Basic Input/Output

- Write a C++ program that accepts user input for their name and age and then displays a personalized greeting.
- Objective: Practice input/output operations using `cin` and `cout`.

```
1 #include<iostream>
2 #include <string>
3 using namespace std;
4
5 main(){
6     string name;
7     int age;
8     cout<<"Enter name: ";
9     getline(cin, name);
10    cout<<"Enter age: ";
11    cin>>age;
12    cout<<"Hello, "<<name<<". You are "<<age<<" years old.";
13 }
```

1.3 POP vs. OOP Comparison Program

- Write two small programs: one using Procedural Programming (POP) to calculate the area of a rectangle, and another using Object-Oriented Programming (OOP) with a class and object for the same task.
- Objective: Highlight the difference between POP and OOP approaches.

```
1 #include<iostream>
2 using namespace std;
3 int recarea(int l, int b){
4     return l*b;
5 }
6 main(){
7     int l, b;
8     cout<<"Enter length: ";
9     cin>>l;
10    cout<<"Enter breadth: ";
11    cin>>b;
12    cout<<"Rectangle area = "<<recarea(l,b)<<endl;
13 }
```

```
1 #include<iostream>
2 using namespace std;
3 struct recarea{
4 public:
5     int l, b;
6     recarea(){
7         cout<<"Enter length: ";
8         cin>>l;
9         cout<<"Enter breadth: ";
10        cin>>b;
11        cout<<"Area of rectangle = "<<l*b<<endl;
12    }
13 };
14 main(){
15     recarea ra;
16 }
```

1.4 Setting Up Development Environment

- Write a program that asks for two numbers and displays their sum. Ensure this is done after setting up the IDE (like Dev C++ or CodeBlocks).
- Objective: Help students understand how to install, configure, and run programs in an IDE.

```
1 #include<iostream>
2 using namespace std;
3 main(){
```

```

4   int a, b;
5   cout<<"Enter first number: ";
6   cin>>a;
7   cout<<"Enter second number";
8   cin>>b;
9   cout<<a<<" + "<<b<<" = "<<a+b<<endl;
10  }

```

2 Variables,DataTypes,andOperators

2.1 VariablesandConstants

- Write a C++ program that demonstrates the use of variables and constants. Create variables of different data types and perform operations on them.
- Objective: Understand the difference between variables and constants.

```

1  #include<iostream>
2  using namespace std;
3  main(){
4      int a = 5;
5      double d = 1.3;
6      char c = 'a';
7      string s = "hello";
8      bool b = 1;
9      const double p = 3.14;
10     const string w = "world";
11     d = p-d;
12     s += ""+w;
13 }

```

2.2 Type Conversion

- Write a C++ program that performs both implicit and explicit type conversions and prints the results.
- Objective: Practice type casting in C++.

```

1  #include <iostream>
2  using namespace std;
3  main() {
4      char c = 'a';
5      int n = ++c;
6      c = (char)n;
7      cout<<n<<end
8      l;
9      cout<<c<<endl

```

2.3 What are the different types of operators in C++? Provide examples of each.

- Write a C++ program that demonstrates arithmetic, relational, logical, and bitwise operators. Perform operations using each type of operator and display the results.
- Objective: Reinforce understanding of different types of operators in C++.

```
1  /*Write a C++ program that demonstrates arithmetic, relational,
2  logical, and bitwise operators. Perform operations using each
3  type of operator and display the results.*/
4  #include<iostream>
5  using namespace std;
6  main(){
7      int a=10, b=5;
8      cout<<a<<" + "<<b<<" = "<<a+b<<endl;
9      cout<<a<<" - "<<b<<" = "<<a-b<<endl;
10     cout<<a<<" * "<<b<<" = "<<a*b<<endl;
11     cout<<a<<" / "<<b<<" = "<<a/b<<endl;
12     cout<<a<<" % "<<b<<" = "<<a%b<<endl;
13     cout<<"++"<<a<<" = "<<++a<<endl;
14     cout<<"--"<<a<<" = "<<--a<<endl;
15     cout<<a<<" == "<<b<<" = "<<(a==b)<<endl;
16     cout<<a<<" != "<<b<<" = "<<(a!=b)<<endl;
17     cout<<a<<" > "<<b<<" = "<<(a>b)<<endl;
18     cout<<a<<" < "<<b<<" = "<<(a<b)<<endl;
19     cout<<a<<" >= "<<b<<" = "<<(a>=b)<<endl;
20     cout<<a<<" <= "<<b<<" = "<<(a<=b)<<endl;
21     cout<<a<<" &&"<<b<<" = "<<(a&&b)<<endl;
22     cout<<a<<" || "<<b<<" = "<<(a||b)<<endl;
23     cout<<"!"<<a<<" = "<<!a<<endl;
24     cout<<a<<" &"<<b<<" = "<<(a&b)<<endl;
25     cout<<a<<" | "<<b<<" = "<<(a|b)<<endl;
26     cout<<a<<" ^ "<<b<<" = "<<(a^b)<<endl;
27     cout<<a<<" <<"<<b<<" = "<<(a<<b)<<endl;
28     cout<<a<<" >>"<<b<<" = "<<(a>>b)<<endl;
29     cout<<"~"<<a<<" = "<<~a<<endl;
30 }
```

3 Control Flow Statements

3.1 Grade Calculator

- Write a C++ program that takes a student's marks as input and calculates the grade based on if-else conditions.
- Objective: Practice conditional statements (if-else).

```
1 #include<iostream>
2 using namespace std;
3 main(){
4     int m;
5     char g;
6     cout<<"Enter student's marks: ";
7     cin>>m;
8     if(m>=90){
9         g='A';
10    }
11    else if (m>=80){
12        g='B';
13    }
14    else if (m>=70){
15        g='C';
16    }
17    else if (m>=60){
18        g='D';
19    }
20    else{
21        g='F';
22    }
23    cout<<"Grade = "<<g<<endl;
24 }
```

3.2 NumberGuessingGame

- Write a C++ program that asks the user to guess a number between 1 and 100. The program should provide hints if the guess is too high or too low. Use loops to allow the user multiple attempts.
- Objective: Understand while loops and conditional logic.

```
1 #include<iostream>
2 using namespace std;
3 main(){
4     srand(time(0));
5     int g, r=(rand()%100)+1;
6     do{
7         cout<<"Guess number between 1 and 100: ";
8         cin>>g;
9         if(g>r){
10            cout<<g<<" is too high"<<endl;
11        }
12        else if (g<r){
13            cout<<g<<" is too low"<<endl;
14        }
15        else {
16            cout<<g<<" is the correct guess"<<endl;
17        }
18    }
```

```

18     }
19     while(g!=r);
20 }

```

3.3 MultiplicationTable

- Write a C++ program to display the multiplication table of a given number using a for loop.
- Objective: Practice using loops.

```

1 #include<iostream>
2 using namespace std;
3 main(){
4     int n, i;
5     cout<<"Enter number: ";
6     cin>>n;
7     for(i=1;i<=10;i++){
8         cout<<n<<" x "<<i<<" = "<<n*i<<endl;
9     }
10 }

```

3.4 NestedControlStructures

- Write a program that prints a right-angled triangle using stars (*) with a nested loop.
- Objective: Learn nested control structures.

```

1 #include<iostream>
2 using namespace std;
3 main(){
4     int i, j;
5     for(i=1;i<=6;i++){
6         for(j=1;j<=i;j++){
7             cout<<"*";
8         }
9         cout<<endl;
10    }
11 }

```

4 Functions and Scope

4.1 Simple Calculator Using Functions

- Write a C++ program that defines functions for basic arithmetic operations (add, subtract, multiply, divide). The main function should call these based on user input.

- Objective: Practice defining and using functions in C++.

```
1 #include<iostream>
2 using namespace std;
3 int add(int a, int b){
4     return a+b;
5 }
6 int sub(int a, int b){
7     return a-b;
8 }
9 int mul(int a, int b){
10    return a*b;
11 }
12 int d(int a, int b){
13    return a/b;
14 }
15 main(){
16     int a,b;
17     char o;
18     cout<<"Enter first number: ";
19     cin>>a;
20     cout<<"Enter second number";
21     cin>>b;
22     cout<<"Enter operation: ";
23     cin>>o;
24     switch(o){
25         case '+':
26             cout<<add(a,b)<<endl;
27             break;
28         case '-':
29             cout<<sub(a,b)<<endl;
30             break;
31         case '*':
32             cout<<mul(a,b)<<endl;
33             break;
34         case '/':
35             cout<<d(a,b)<<endl;
36             break;
37     }
38 }
```

4.2 Factorial Calculation Using Recursion

- Write a C++ program that calculates the factorial of a number using recursion.
- Objective: Understand recursion in functions.

```
1 #include<iostream>
2 using namespace std;
3 int f(int n){
```

```

4     if (n==1){
5         return 1;
6     }
7     else{
8         return (n*f(n-1));
9     }
10 }
11 main(){
12     int    n;
13     cout<<"Enter number: ";
14     cin>>n;
15     cout<<n<<"! = "<<f(n);
16 }

```

4.3 VariableScope

- Write a program that demonstrates the difference between local and global variables in C++. Use functions to show scope.
- Objective: Reinforce the concept of variable scope.

```

1 #include<iostream>
2 using namespace std;
3 int g=5;
4 void func1(){
5     int l=3;
6     cout<<"local variable: "<<l<<endl;
7     cout<<"global variable: "<<g<<endl;
8 }
9 main(){
10     func1();
11     cout<<"local variable: "<<l<<endl;    //gives error
12     int l=7;
13     cout<<"local variable: "<<l<<endl;
14     cout<<"global variable: "<<g<<endl;
15 }

```

5 ArraysandStrings

5.1 ArraySumandAverage

- Write a C++ program that accepts an array of integers, calculates the sum and average, and displays the results.
- Objective: Understand basic array manipulation.

```

1 #include<iostream>
2 using namespace std;

```



```

3 main(){
4     int    a[5], i, s=0;
5     for(i=0;i<5;i++){
6         cout<<"Enter "<<i<<"th element: ";
7         cin>>a[i];
8         s+=a[i];
9     }
10    cout<<"sum = "<<s<<endl;
11    cout<<"average = "<<s/5<<endl;
12 }

```

5.2 MatrixAddition

- Write a C++ program to perform matrix addition on two 2x2 matrices.
- Objective: Practice multi-dimensional arrays.

```

1 #include<iostream>
2 using namespace std;
3 main(){
4     int a[2][2], b[2][2], i, j;
5     for(i=0;i<2;i++){
6         for(j=0;j<2;j++){
7             cout<<"Enter ["<<i+1<<"]["<<j+1<<"]    elementof 1st
8                 matrix: ";
9             cin>>a[i][j];
10        }
11    }
12    for(i=0;i<2;i++){
13        for(j=0;j<2;j++){
14            cout<<"Enter ["<<i+1<<"]["<<j+1<<"]    elementof 2nd
15                matrix: ";
16            cin>>b[i][j];
17        }
18    }
19    cout<<"sum = "<<endl;
20    for(i=0;i<2;i++){
21        for(j=0;j<2;j++){
22            cout<<a[i][j]+b[i][j]<<"    ";
23        }
24        cout<<endl;
25    }
26 }

```

5.3 String Palindrome Check

- Write a C++ program to check if a given string is a palindrome (reads the same forwards and backwards).
- Objective: Practice string operations.

```

1 #include<iostream>
2 #include<string>
3 #include<cmath>
4 using namespace std;
5 main(){
6     string s;
7     cout<<"Enter string: ";
8     cin>>s;
9     int i, j, l=s.length()-1;
10    for(i=0, j=l; i<l, j>=0; i++, j--){
11        if(s[i] != s[j]){
12            cout<<s<<" is not a palindrome";
13            break;
14        }
15    }
16    if(i==l+1){
17        cout<<s<<" is a palindrome";
18    }
19 }

```

6 Introduction to Object-Oriented Programming

6.1 Class for a Simple Calculator

- Write a C++ program that defines a class Calculator with functions for addition, subtraction, multiplication, and division. Create objects to use these functions.
- Objective: Introduce basic class structure.

```

1 #include<iostream>
2 using namespace std;
3 Calculator{
4 public :
5     int addition(int a, int b){
6         return a+b;
7     }
8     int subtraction(int a, int b){
9         return a-b;
10    }
11    int multiplication(int a, int b){
12        return a*b;
13    }
14    int division(int a, int b){
15        return a/b;
16    }
17 };
18 main(){
19     int a, b;
20     cout<<"Enter 1st number: ";

```

```

21     cin>>a;
22     cout<<"Enter 2nd number";
23     cin>>b;
24     Calculator o;
25     cout<<a<<" + "<<b<<" = "<<o.addition(a,b)<<endl;
26     cout<<a<<" - "<<b<<" = "<<o.subtraction(a,b)<<endl;
27     cout<<a<<" * "<<b<<" = "<<o.multiplication(a,b)<<endl;
28     cout<<a<<" / "<<b<<" = "<<o.division(a,b)<<endl;
29 }

```

6.2 Class for Bank Account

- Create a class BankAccount with data members like balance and member functions like deposit and withdraw. Implement encapsulation by keeping the data members private.
- Objective: Understand encapsulation in classes

```

1  #include<iostream>
2  using namespace std;
3  BankAccount{
4  class private:
5      int balance;
6  public:
7      void deposit(int b){
8          int d;
9          cout<<"Enter amount to deposit: ";
10         cin>>d;
11         balance=b;
12         cout<<"Balance after deposit: "<<balance+d<<endl;
13     }
14     void withdraw(int b){
15         int w;
16         cout<<"Enter amount to withdraw: ";
17         cin>>w;
18         balance=b;
19         cout<<"Balance after withdraw: "<<balance-w<<endl;
20     }
21 };
22 main(){
23     int b, ch;
24     cout<<"Enter current balance: ";
25     cin>>b;
26     BankAccount o;
27     cout<<"1. Deposit"<<endl;
28     cout<<"2. Withdraw"<<endl;
29     cout<<"3. Exit"<<endl;
30     cout<<"Enter your choice: ";
31     cin>>ch;
32     switch(ch){

```

```

33     case 1:
34         o.deposit(b);
35         break;
36     case 2:
37         o.withdraw(b);
38         break;
39     case 3:
40         cout<<"Thank you"<<endl;
41         break;
42     default:
43         cout<<"Invalid input"<<endl;
44         break;
45 }
46 }

```

6.3 Inheritance Example

- Write a program that implements inheritance using a base class Person and derived classes Student and Teacher. Demonstrate reusability through inheritance.
- Objective: Learn the concept of inheritance.

```

1  #include<iostream>
2  using namespace std;
3  Person{
4  public:
5      string n;
6      int a;
7      void inP(string t){
8          cout<<"Enter "<<t<<" name:";
9          cin>>n;
10         cout<<"Enter "<<t<<" age: ";
11         cin>>a;
12     }
13 };
14 class Student: public Person{
15     public:
16         int c;
17         string t="student's";
18         void inS(){
19             inP(t);
20             cout<<"Enter student's class: ";
21             cin>>c;
22         }
23         void outS(){
24             cout<<"Student's name:"<<n<<endl;
25             cout<<"Student's age: "<<a<<endl;
26             cout<<"Student's class: "<<c<<endl;
27         }
28 };

```

```
29 class Teacher: public Person{
30     public:
31         string s, t="teacher's";
32         void inT(){
33             inP(t);
34             cout<<"Enter teacher's subject: ";
35             cin>>s;
36         }
37         void outT(){
38             cout<<"Teacher's name:"<<n<<endl;
39             cout<<"Teacher's age: "<<a<<endl;
40             cout<<"Teacher's subject: "<<s<<endl;
41         }
42 };
43 main(){
44     Student o1;
45     Teacher o2;
46     o1.inS();
47     o2.inT();
48     o1.outS();
49     o2.outT();
50 }
```