

Practice Question & Answers for Programming Fundamentals.

Introduction to C++

C++ is a powerful, high-performance programming language widely used in various fields of technology. Its flexibility allows it to be used in system programming, game development, device drivers, and real-time systems. For example:

- **Samsung Gear Watch OS** is built with C++ for optimal performance.
- **Drivers for hardware devices** (like printers, graphics cards) are often written in C++ to interact with the hardware efficiently.
- **Operating systems**, such as parts of Windows and macOS, are built using C++ for performance-critical components.
- **Game development**: Popular game engines like Unreal Engine are built using C++ due to its performance and control over system resources.
- **Financial trading systems**: Speed is critical, and C++ helps in building fast, reliable trading algorithms.
- **Embedded systems**: C++ is used in car systems, medical devices, and smart home devices because of its efficiency.

With C++, you can pursue careers in fields like system architecture, game development, embedded systems, and even financial systems, which are all high-demand fields with lucrative job opportunities.

Data Types & Variables

Practice Questions:

1. Write a program to store a student's name, age, and grade, and display the values.
2. Create a program that calculates the perimeter and area of a rectangle using integer variables.
3. Declare variables for height (float), name (string), and age (int). Print them.

4. Create a program to convert temperature from Fahrenheit to Celsius.
5. Write a program to calculate the average of three numbers entered by the user.

Additional Questions with Solutions:

1. Calculate Circle Area and Circumference:

- Formula for area: πr^2
- Formula for circumference: $2\pi r$

```
#include <iostream>
#define PI 3.14159
using namespace std;

int main() {
    float radius, area, circumference;
    cout << "Enter radius: ";
    cin >> radius;
    area = PI * radius * radius;
    circumference = 2 * PI * radius;
    cout << "Area: " << area << ", Circumference: " << circumference;
    return 0;
}
```

2. Find Largest of Two Numbers:

- Use two variables and an `if` statement to compare them.

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

int main() {
    int num1, num2;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    if (num1 > num2) {
        cout << "Largest: " << num1;
    } else {
        cout << "Largest: " << num2;
    }
}
```

```
    ... return 0;
}
```

Console Input

Practice Questions:

1. Create a program that takes a user's age and name, then greets them.
2. Write a program that asks for two numbers and prints their sum, product, and difference.
3. Develop a program that asks for three sides of a triangle and calculates its perimeter.
4. Ask for a number and check if it's even or odd.
5. Write a program that takes a number and displays its square and cube.

Additional Questions with Solutions:

1. Square Root Calculation:

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

int main() {
    float num;
    cout << "Enter a number: ";
    cin >> num;
    cout << "Square root: " << sqrt(num);
    return 0;
}
```

2. Convert Hours to Minutes:

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

int main() {
    int hours;
    cout << "Enter hours: ";
    cin >> hours;
```

```
.... cout << "Minutes: " << hours * 60;
.... return 0;
}
```

Operators in C++ & Math Functions

Practice Questions:

1. Write a program to calculate the modulus of two numbers.
2. Calculate the power of a number (use `pow()` from `<cmath>`).
3. Write a program that performs addition, subtraction, multiplication, and division on two numbers.
4. Use relational operators to compare two numbers and print which one is larger.
5. Develop a program to calculate the distance between two points using the distance formula.

Additional Questions with Solutions:

1. Simple Calculator:

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

int main() {
    float num1, num2;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    cout << "Sum: " << num1 + num2 << "\n";
    cout << "Difference: " << num1 - num2 << "\n";
    cout << "Product: " << num1 * num2 << "\n";
    cout << "Quotient: " << num1 / num2 << "\n";
    return 0;
}
```

2. Compound Interest Calculation:

```
#include <iostream>
#include <cmath>
```

```
using namespace std;

int main() {
    double principal, rate, time, compound_interest;
    cout << "Enter principal, rate, and time: ";
    cin >> principal >> rate >> time;
    compound_interest = principal * pow((1 + rate / 100), time);
    cout << "Compound interest: " << compound_interest;
    return 0;
}
```

Loops

Practice Questions:

1. Write a program that prints numbers from 1 to 100 using a loop.
2. Develop a program that calculates the sum of all even numbers from 1 to 50.
3. Create a program that prints the multiplication table of a number entered by the user.
4. Write a program that calculates the factorial of a number using a loop.
5. Print the Fibonacci series up to a given number.

Additional Questions with Solutions:

1. Print Numbers in Reverse Order:

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

int main() {
    for (int i = 100; i >= 1; i--) {
        cout << i << " ";
    }
    return 0;
}
```

2. Sum of Digits:

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

int main() {
    int num, sum = 0, digit;
    cout << "Enter a number: ";
    cin >> num;
    while (num > 0) {
        digit = num % 10;
        sum += digit;
        num /= 10;
    }
    cout << "Sum of digits: " << sum;
    return 0;
}
```

Selection Statements

Practice Questions:

1. Write a program to check if a number is positive, negative, or zero using `if-else`.
2. Create a program that checks if a number is even or odd using a `switch` statement.
3. Write a program that takes a day number (1-7) and prints the corresponding day using a `switch` statement.
4. Use nested `if-else` to find the largest of three numbers.
5. Write a program that categorizes a person based on their age.

Additional Questions with Solutions:

1. Check Voting Eligibility:

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

int main() {
    int age;
    cout << "Enter your age: ";
    cin >> age;
```

```
.... if (age >= 18) {  
..... cout << "You are eligible to vote.";  
} else {  
..... cout << "You are not eligible to vote."  
}  
return 0;  
}
```

2. Grade Calculation:

```
#include <iostream>  
using namespace std;  
  
int main() {  
.... int marks;  
.... cout << "Enter your marks: ";  
.... cin >> marks;  
.... if (marks >= 90) {  
..... cout << "Grade: A";  
} else if (marks >= 75) {  
..... cout << "Grade: B";  
} else if (marks >= 50) {  
..... cout << "Grade: C";  
} else {  
..... cout << "Fail";  
}  
.... return 0;  
}
```

Functions

Practice Questions:

1. Write a function that takes two numbers and returns their sum.
2. Create a program with a function to check whether a number is prime.
3. Write a function that calculates the factorial of a number.
4. Develop a program with a function that returns the largest of two numbers.
5. Create a function that calculates the GCD of two numbers.

Additional Questions with Solutions:

1. Prime Check Function:

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

bool isPrime(int n) {
    if (n <= 1) return false;
    for (int i = 2; i < n; i++) {
        if (n % i == 0) return false;
    }
    return true;
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    if (isPrime(num)) {
        cout << num << " is a prime number.";
    } else {
        cout << num << " is not
a prime number.";
    }
    return 0;
}
```

2. **Greatest of Three Numbers Function**:

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

int largest(int a, int b, int c) {
 if (a >= b && a >= c) return a;
 else if (b >= a && b >= c) return b;
 else return c;
}
```

```
int main() {
 int x, y, z;
 cout << "Enter three numbers: ";
 cin >> x >> y >> z;
 cout << "Largest: " << largest(x, y, z);
 return 0;
}
```

---

## Arrays

### Practice Questions:

1. Write a program that calculates the sum of all elements in an array of size 10.
2. Create a program that finds the largest and smallest elements in an array.
3. Write a program that takes an array of 5 integers and sorts them in ascending order.
4. Develop a program that calculates the average of elements in an array.
5. Write a program that checks whether a given number exists in an array.

### Additional Questions with Solutions:

#### 1. Find Maximum Element in an Array:

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

int main() {
 int arr[5], max;
 cout << "Enter 5 elements: ";
 for (int i = 0; i < 5; i++) {
 cin >> arr[i];
 }
 max = arr[0];
 for (int i = 1; i < 5; i++) {
 if (arr[i] > max) max = arr[i];
 }
 cout << "Maximum: " << max;
 return 0;
}
```

## 2. Reverse Array:

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

int main() {
 int arr[5], temp;
 cout << "Enter 5 elements: ";
 for (int i = 0; i < 5; i++) {
 cin >> arr[i];
 }
 for (int i = 0; i < 5 / 2; i++) {
 temp = arr[i];
 arr[i] = arr[4 - i];
 arr[4 - i] = temp;
 }
 cout << "Reversed array: ";
 for (int i = 0; i < 5; i++) {
 cout << arr[i] << " ";
 }
 return 0;
}
```

---

## Structures

### Practice Questions:

1. Write a program to define a `struct` for a student with fields for name, age, and grade.
2. Create a program that takes input for three employees (name, ID, salary) using a `struct` and displays the data.
3. Define a `struct` for a book with fields for title, author, and price, and create an array of 3 books.
4. Write a program to store and display information for 3 cars (brand, model, year) using a `struct`.
5. Create a program that stores a person's details (name, age, address) using a `struct`.

### Additional Questions with Solutions:

## 1. Structure for Student Information:

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

struct Student {
 string name;
 int age;
 float grade;
};

int main() {
 Student s;
 cout << "Enter name: ";
 cin >> s.name;
 cout << "Enter age: ";
 cin >> s.age;
 cout << "Enter grade: ";
 cin >> s.grade;
 cout << "Student: " << s.name << ", Age: " << s.age << ", Grade: " << s.gr
 return 0;
}
```



## 2. Store Employee Information:

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

struct Employee {
 string name;
 int id;
 float salary;
};

int main() {
 Employee e[3];
 for (int i = 0; i < 3; i++) {
 cout << "Enter name, ID, and salary of employee " << i + 1 << ": ";
 cin >> e[i].name >> e[i].id >> e[i].salary;
 }
 cout << "\nEmployee Details:\n";
}
```

```
.... for (int i = 0; i < 3; i++) {
.... cout << e[i].name << " | " << e[i].id << " | " << e[i].salary << "\n";
.... }
.... return 0;
`
```

---

## Pointers

### Practice Questions:

1. Write a program to declare an integer variable, store its address in a pointer, and then display the value of the variable using the pointer.
2. Create a program that swaps two numbers using pointers.
3. Write a program that demonstrates pointer arithmetic by incrementing and decrementing a pointer to an array.
4. Develop a program that dynamically allocates memory for an array of integers using `new`, takes input for the array, and then deallocates the memory using `delete[]`.
5. Create a program that uses a pointer to access and modify the value of a variable from a different function.

### Additional Questions with Solutions:

#### 1. Swap Two Numbers Using Pointers:

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

void swap(int *x, int *y) {
 int temp = *x;
 *x = *y;
 *y = temp;
}

int main() {
 int a = 10, b = 20;
 cout << "Before swap: " << a << " " << b << "\n";
 swap(&a, &b);
 cout << "After swap: " << a << " " << b;
```

```
 ... return 0;
}
```

## 2. Pointer Arithmetic:

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

int main() {
 ... int arr[] = {10, 20, 30, 40, 50};
 int *ptr = arr;
 ...

 cout << "First element: " << *ptr << "\n";
 ptr++;
 cout << "Second element: " << *ptr << "\n";
 ptr--;
 cout << "First element again: " << *ptr << "\n";
 return 0;
}
```

---

## Real-World Application with Solution: ATM System

**Task:** Write an ATM program that can store and withdraw cash for three customers, each with a name, account number, and balance.

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

struct Customer {
 string name;
 int accountNumber;
 float balance;
};

void deposit(Customer &cust, float amount) {
 cust.balance += amount;
 cout << amount << " deposited. New balance: " << cust.balance << "\n";
}
```

```
void withdraw(Customer &cust, float amount) {
 if (amount > cust.balance) {
 cout << "Insufficient balance.\n";
 } else {
 cust.balance -= amount;
 cout << amount << " withdrawn. New balance: " << cust.balance << "\n";
 }
}

int main() {
 Customer customers[3] = {
 {"Alice", 12345, 5000.0},
 {"Bob", 67890, 3000.0},
 {"Charlie", 13579, 10000.0}
 };

 int choice, accNum;
 float amount;
 cout << "Welcome to the ATM\n";

 cout << "Enter account number: ";
 cin >> accNum;

 Customer *currentCustomer = nullptr;
 for (int i = 0; i < 3; i++) {
 if (customers[i].accountNumber == accNum) {
 currentCustomer = &customers[i];
 break;
 }
 }

 if (currentCustomer == nullptr) {
 cout << "Invalid account number.\n";
 return 0;
 }

 while (true) {
 cout << "1. Deposit\n2. Withdraw\n3. Exit\nChoose an option: ";
 cin >> choice;
 if (choice == 3) break;

 cout << "Enter amount: ";
 cin >> amount;
 }
}
```

```
 switch (choice) {
 case 1:
 deposit(*currentCustomer, amount);
 break;
 case 2:
 withdraw(*currentCustomer, amount);
 break;
 default:
 cout << "Invalid option.\n";
 }
 }

 return 0;
}
```

## Recursion

Recursion is a technique where a function calls itself to solve a smaller instance of the problem until a base condition is met. It is useful in solving problems that can be divided into similar sub-problems.

### Real-World Application of Recursion

Recursion is often used in tasks such as:

- **File Systems:** Traversing directories.
- **Mathematical Computations:** Calculating factorials, Fibonacci sequences.
- **Sorting Algorithms:** Quick sort and merge sort.
- **Graphics:** Fractal generation.

### Practice Questions:

1. Write a recursive function to calculate the factorial of a number.
2. Create a recursive function to calculate the nth Fibonacci number.
3. Write a program that uses recursion to find the greatest common divisor (GCD) of two numbers.
4. Develop a recursive function to calculate the sum of digits of a number.

## 5. Implement a recursive function that reverses a string.

### Additional Questions with Solutions:

#### 1. Factorial Calculation:

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

int factorial(int n) {
 if (n == 0) return 1;
 return n * factorial(n - 1);
}

int main() {
 int num;
 cout << "Enter a number: ";
 cin >> num;
 cout << "Factorial: " << factorial(num);
 return 0;
}
```

#### 2. Fibonacci Sequence:

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

int fibonacci(int n) {
 if (n <= 1) return n;
 return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
 int num;
 cout << "Enter a number: ";
 cin >> num;
 cout << "Fibonacci: " << fibonacci(num);
 return 0;
}
```

## Recursion Practice Questions:

1. Write a recursive function to find the sum of all elements in an array.
2. Develop a recursive function that counts how many times a specific character appears in a string.
3. Create a program using recursion to calculate the power of a number (e.g.,  $a^b$  ).