SOLUTION BOOK

❤️ **From SIDDHARTH SINGH**

BASIC

1) **"Hello, World!" Program**

CODE:

```
#include <iostream>

int main() {
   std::cout << "Hello World!";
   return 0;
}
```

Output:

```
Hello World!
```

2) **Print Number Entered by User**

CODE:

```
#include <iostream>
using namespace std;
int main() {
   int number;

   cout << "Enter an integer: ";
   cin >> number;

   cout << "You entered " << number;
   return 0;
}
```

Output:

```
Enter an integer: 121
You entered 121
```

3) Add Two Numbers

NOTE: // means Comment that means that line will not be executed by compiler

CODE:

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

int main()
{
    int firstNumber, secondNumber, sumOfTwoNumbers;

    cout << "Enter two integers: ";
    cin >> firstNumber >> secondNumber;

    // sum of two numbers in stored in variable sumOfTwoNumbers
    sumOfTwoNumbers = firstNumber + secondNumber;

    // Prints sum
    cout << firstNumber << " + " <<  secondNumber << " = " << sumOfTwoNumbers;

    return 0;
}
```

Output:

```
Enter two integers: 12
9
12 + 9 = 21
```

4) Find Quotient and Remainder

CODE:

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

int main()
{
    int divisor, dividend, quotient, remainder;

    cout << "Enter dividend: ";
    cin >> dividend;

    cout << "Enter divisor: ";
    cin >> divisor;
    quotient = dividend / divisor;
    remainder = dividend % divisor;

    cout << "Quotient = " << quotient << endl;
    cout << "Remainder = " << remainder;

    return 0;
}
```

Output

```
Enter dividend: 13
Enter divisor: 4
Quotient = 3
Remainder = 1
```

CONCEPT:
The division operator / computes the quotient (either between float or integer variables).

The modulus operator % computes the remainder when one integer is divided by another (modulus operator cannot be used for floating-type variables).

5) Find Size of int, float, double and char in your Computer

CODE:

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

int main()
{
    cout << "Size of char: " << sizeof(char) << " byte" << endl;
    cout << "Size of int: " << sizeof(int) << " bytes" << endl;
    cout << "Size of float: " << sizeof(float) << " bytes" << endl;
    cout << "Size of double: " << sizeof(double) << " bytes" << endl;

    return 0;
}
```

Output

```
Size of char: 1 byte
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
```

Note: You may get different results if you are using an old computer.

CONCEPT:

To find the size of the variable, sizeof operator is used.

sizeof(dataType);

6) Swap Two Numbers

Method 1: Swap Numbers (Using Temporary Variable)

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

int main()
{
    int a = 5, b = 10, temp;

    cout << "Before swapping." << endl;
    cout << "a = " << a << ", b = " << b << endl;

    temp = a;
    a = b;
    b = temp;

    cout << "\nAfter swapping." << endl;
    cout << "a = " << a << ", b = " << b << endl;

    return 0;
}
```

Output

```
Before swapping.
a = 5, b = 10

After swapping.
a = 10, b = 5
```

Method 2: Swap Numbers Without Using Temporary Variables

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

int main()
```

```
{

    int a = 5, b = 10;

    cout << "Before swapping." << endl;
    cout << "a = " << a << ", b = " << b << endl;

    a = a + b;
    b = a - b;
    a = a - b;

    cout << "\nAfter swapping." << endl;
    cout << "a = " << a << ", b = " << b << endl;

    return 0;
}
```

CONCEPT:

Initially, a = 5 and b = 10.
Then, we add a and b and store it in a with the code a = a + b. This means a = 5 + 10. So, a = 15 now.
Then we use the code b = a - b. This means b = 15 - 10. So, b = 5 now.
Again, we use the code a = a - b. This means a = 15 - 5. So finally, a = 10.
Hence, the numbers have been swapped.

Note: We can also use multiplication and division instead of addition and subtraction. However, this won't work if one of the numbers is 0.

```
int a = 5, b = 10;

// using multiplication and division for swapping
a = a * b;    // a = 50
b = a / b;    // b = 5
a = a / b;    // a = 10
```

7) Program to Find ASCII Value of a Character

THEORY:

A character variable holds ASCII value (an integer number between 0 and 127) rather than that character itself in C programming. That value is known as ASCII value.

For example, the ASCII value of 'A' is 65. What this means is that, if you assign 'A' to a character variable, 65 is stored in that variable rather than 'A' itself.

CODE:

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

int main() {
 char c;
 cout << "Enter a character: ";
 cin >> c;
 cout << "ASCII Value of " << c << " is " << int(c);
 return 0;
}
```

Output

```
Enter a character: p
ASCII Value of p is 112
```

CONCEPT:

When we explicitly print the integer value of a char type, it's corresponding ASCII value is printed.

8) Program to Multiply two Numbers

CODE:

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

int main() {
  double num1, num2, product;
  cout << "Enter two numbers: ";

  // stores two floating point numbers in num1 and num2 respectively
  cin >> num1 >> num2;

  // performs multiplication and stores the result in product variable
  product = num1 * num2;

  cout << "Product = " << product;

  return 0;
}
```

Output

```
Enter two numbers: 3.4
5.5
Product = 18.7
```