

1. Swapping two numbers without using a third variable:

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

class Swap
{
    int a, b;

public:
    void getdata()
    {
        cout << "Enter the value of a and b: ";
        cin >> a >> b;
    }
    void swap()
    {
        a = a + b;
        b = a - b;
        a = a - b;
    }
    void display()
    {
        cout << "After swapping the value of a and b is: " << a << " "
<< b;
    }
};

int main()
{
    Swap s;
    s.getdata();
    s.swap();
    s.display();
    return 0;
}
```

• Output:

Enter the value of a and b: 10 20

Before swapping: a = 10 and b = 20

After swapping: a = 20 and b = 10

2. Printing the first three powers of an integer:

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

class power
{
    int n;

public:
    void getdata()
    {
        cout << "Enter the number: ";
        cin >> n;
    }
    void display()
    {
        cout << "The first three powers of " << n << " are: " << endl;
        cout << n << "^1 = " << n << endl;
        cout << n << "^2 = " << n * n << endl;
        cout << n << "^3 = " << n * n * n << endl;
    }
};

int main()
{
    power p;
    p.getdata();
    p.display();
    return 0;
}
```

• Output:

Enter the number: 5

The first three powers of 5 are:

5¹ = 5
5² = 25
5³ = 125

3. Identifying the type of triangle based on its sides:

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

class triangle
{
    int a, b, c;

public:
    void getdata()
    {
        cout << "Enter the three sides of triangle: ";
        cin >> a >> b >> c;
    }
    void check()
    {
        if (a == b && b == c)
        {
            cout << "Triangle is equilateral";
        }
        else if (a == b || b == c || c == a)
        {
            cout << "Triangle is isosceles";
        }
        else
        {
            cout << "Triangle is scalene";
        }
    }
};

int main()
{
    triangle t;
    t.getdata();
    t.check();
    return 0;
}
```

• Output:

```
Enter the three sides of triangle: 10 30 30
```

```
Triangle is isosceles
```

4. Reversing a number:

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

class Reverse
{
    int num, rev = 0, rem;

public:
    void getdata()
    {
        cout << "Enter the number: ";
        cin >> num;
    }
    void reverse()
    {
        while (num != 0)
        {
            rem = num % 10;
            rev = rev * 10 + rem;
            num /= 10;
        }
    }
    void display()
    {
        cout << "The reverse of the number is: " << rev;
    }
};

int main()
{
    Reverse r;
    r.getdata();
    r.reverse();
    r.display();
    return 0;
}
```

• Output:

Enter the number: 12345

The reverse of the number is: 54321

5. Printing Fibonacci series:

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

class Fibonacci
{
    int n;

public:
    void getdata()
    {
        cout << "Enter the limit of the series: ";
        cin >> n;
    }
    void generate()
    {
        int a = 0, b = 1, c;
        cout << "The Fibonacci series is: " << a << " " << b << " ";
        for (int i = 0; i < n - 2; i++)
        {
            c = a + b;
            cout << c << " ";
            a = b;
            b = c;
        }
    }
};

int main()
{
    Fibonacci f;
    f.getdata();
    f.generate();
    return 0;
}
```

• Output:

Enter the limit of the series: 10

The Fibonacci series is: 0 1 1 2 3 5 8 13 21 34

6. Printing prime numbers up to 300:

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

class Prime
{
    int i, j;

public:
    void prime()
    {
        for (i = 1; i <= 300; i++)
        {
            int count = 0;
            for (j = 1; j <= i; j++)
            {
                if (i % j == 0)
                {
                    count++;
                }
            }
            if (count == 2)
            {
                cout << i << " ";
            }
        }
    }
};

int main()
{
    Prime p;
    p.prime();
    return 0;
}
```

• Output:

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191
193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283
293
```

7. Checking if a number is a palindrome:

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

class Palindrome
{
    int num, rev = 0, rem;

public:
    void getdata()
    {
        cout << "Enter the number: ";
        cin >> num;
    }
    void check()
    {
        int temp = num;
        while (num > 0)
        {
            rem = num % 10;
            rev = rev * 10 + rem;
            num = num / 10;
        }
        if (temp == rev)
        {
            cout << "The number is palindrome";
        }
        else
        {
            cout << "The number is not palindrome";
        }
    }
};

int main()
{
    Palindrome p;
    p.getdata();
    p.check();
    return 0;
}
```

• Output:

Enter the number: 12321

The number is palindrome

8. Abbreviating a name using arrays:

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

class name
{
    char first[20], last[20];

public:
    void getdata()
    {
        cout << "Enter first name: ";
        cin >> first;
        cout << "Enter last name: ";
        cin >> last;
    }
    void display()
    {
        cout << "Abbreviated name: " << first[0] << ". " << last[0] <<
        ".";
    }
};

int main()
{
    name n;
    n.getdata();
    n.display();
    return 0;
}
```

• Output:

Enter first name: Musheer

Enter last name: Alam

Abbreviated name: M. A.

9. Converting number to character:

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

class NumberToCharacter
{
    int number;

public:
    void getNumber()
    {
        cout << "Enter a number: ";
        cin >> number;
    }

    void convertNumberToCharacter()
    {
        cout << "The character form of " << number << " is " <<
char(number) << endl;
    }
};

int main()
{
    NumberToCharacter obj;
    obj.getNumber();
    obj.convertNumberToCharacter();
    return 0;
}
```

• Output:

Enter a number: 65

The character form of 65 is A

10. Converting decimal to binary:

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

class DecimalToBinary
{
    int num;

public:
    void getNum()
    {
        cout << "Enter a decimal number: ";
        cin >> num;
    }

    void convert()
    {
        int binary[32];
        int i = 0;

        while (num > 0)
        {
            binary[i] = num % 2;
            num = num / 2;
            i++;
        }

        cout << "Binary number: ";
        for (int j = i - 1; j >= 0; j--)
        {
            cout << binary[j];
        }
    }
};

int main()
{
    DecimalToBinary obj;
    obj.getNum();
    obj.convert();
    return 0;
}
```

• Output:

```
Enter a decimal number: 12

Binary number: 1100
```

11. Finding factorial using a recursive function:

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

class factorial
{
    int n;

public:
    void getdata()
    {
        cout << "Enter the number: ";
        cin >> n;
    }
    int fact(int n)
    {
        if (n == 0)
            return 1;
        else
            return n * fact(n - 1);
    }
    void display()
    {
        cout << "Factorial of " << n << " is " << fact(n) << endl;
    }
};

int main()
{
    factorial f;
    f.getdata();
    f.display();
    return 0;
}
```

• Output:

```
Enter the number: 5
Factorial of 5 is 120
```

12. Using call by value and call by reference functions:

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

void callbyvalue(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
    cout << "After swapping using call by value: " << endl;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
}

void callbyreference(int &x, int &y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
    cout << "After swapping using call by reference: " << endl;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
}

int main()
{
    int a, b;
    cout << "Enter the value of a: ";
    cin >> a;
    cout << "Enter the value of b: ";
    cin >> b;
    callbyvalue(a, b);
    callbyreference(a, b);
    return 0;
}
```

• Output:

After swapping using call by value:

x = 20

y = 10

After swapping using call by reference:

x = 20

y = 10

13. Printing the alphabet triangle:

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

int main()
{
    int i, j, k, size;
    char ch;

    cout << "Enter the size of the triangle: ";
    cin >> size;

    for (i = 1; i <= size; i++)
    {
        ch = 'A';
        for (j = size; j >= i; j--)
        {
            cout << " ";
        }
        for (k = 1; k <= i; k++)
        {
            cout << ch;
            ch++;
        }
        ch--;
        for (k = 1; k < i; k++)
        {
            ch--;
            cout << ch;
        }
        cout << endl;
    }
    return 0;
}
```

• Output:

Enter the size of the triangle: 3

```
A
ABA
ABCBA
```

14. Calculating the dimensions of a circle using functions:

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

class circle
{
    float r;

public:
    void getdata()
    {
        cout << "Enter the radius of the circle: ";
        cin >> r;
    }
    void area()
    {
        cout << "Area of the circle is: " << 3.14 * r * r << endl;
    }
    void circumference()
    {
        cout << "Circumference of the circle is: " << 2 * 3.14 * r <<
endl;
    }
};

class cylinder
{
    float r, h;

public:
    void getdata()
    {
        cout << "Enter the radius of the cylinder: ";
        cin >> r;
        cout << "Enter the height of the cylinder: ";
        cin >> h;
    }
    void volume()
    {
        cout << "Volume of the cylinder is: " << 3.14 * r * r * h <<
endl;
    }
    void surface_area()
    {
        cout << "Surface area of the cylinder is: " << 2 * 3.14 * r * h
+ 2 * 3.14 * r * r << endl;
    }
};
```

```

int main()
{
    int ch;
    circle c;
    cylinder cy;
    do
    {
        cout << "1. Area of circle\n2. Circumference of circle\n3.
Volume of cylinder\n4. Surface area of cylinder\n5. Exit\nEnter your
choice: ";
        cin >> ch;
        switch (ch)
        {
            case 1:
                c.getdata();
                c.area();
                break;
            case 2:
                c.getdata();
                c.circumference();
                break;
            case 3:
                cy.getdata();
                cy.volume();
                break;
            case 4:
                cy.getdata();
                cy.surface_area();
                break;
            case 5:
                break;
            default:
                cout << "Invalid choice\n";
        }
    } while (ch != 5);
    return 0;
}

```

• Output:

```

1. Area of circle
2. Circumference of circle
3. Volume of cylinder
4. Surface area of cylinder
5. Exit
Enter your choice: 1
Enter the radius of the circle: 5
Area of the circle is: 78.5

```