```
1. #include <iostream>
using namespace std;
int main()
{
   cout<<"Enter the size of array : ";
   int n;cin>>n;
   int arr[n];
   cout<<"Enter elements :";
   for(int i=0;i<n;i++){
      cin>>arr[i];
   }
   for(int i=0;i<n-1;i++){
      for(int j=i+1;j<n;j++){
         if(arr[i]>arr[j]){
            int tmp=arr[i];
            arr[i]=arr[j];
            arr[j]=tmp;
         }
      }
   }
   cout<<"The ascending array is :";
   for(int i=0;i<n;i++){
      cout<<arr[i]<<" ";
   }
   return 0;
}
```

2.
```
import java.util.Scanner;

public class Descending_sort {

  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the number of elements :");
    int n = input.nextInt();

    int[] arr = new int[n];
    System.out.print("Enter the elements :");
    for (int i = 0; i < n; i++) {
      arr[i] = input.nextInt();
    }

    descendingSort(arr);

    // Print the sorted array
    System.out.print("The Array in descending order is :");
    for (int num : arr) {
      System.out.print(num + " ");
    }
    input.close();
  }

  public static void descendingSort(int[] arr) {
    int n = arr.length;
```

```
    for (int i = 0; i < n - 1; i++) {
      for (int j = i+1; j< n; j++) {
        if (arr[i] < arr[j]) {
          // Swap the elements
          int tmp = arr[i];
          arr[i] = arr[j];
          arr[j] = tmp;
        }
      }
    }
  }
}
```

3.

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

int main()
{
    int rows;

    // Get the number of rows from the user
    cout << "Enter the number of rows: ";
    cin >> rows;

    // Outer loop for rows
    for (int i = 1; i <= rows; i++)
```

```
    {
        // Inner loop for columns
        for (int j = 1; j <= i; j++)
        {
            cout << "* ";
        }

        cout << endl;
    }

    return 0;
}
```

4.
```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
   ofstream file("test.txt");
    if (!file.is_open())
    {
        cout << "Error opening the file." << endl;
        return 1;
    }
    string name;
    int rollNumber;
    cout << "Enter your name: ";
```

```
    getline(cin, name);
    cout << "Enter your roll number: ";
    cin >> rollNumber;
    file << "Name: " << name << endl;
    file << "Roll Number: " << rollNumber << endl;
    file.close();
    cout << "Data written to the file successfully." << endl;

    return 0;
}
```

5.
```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
   ifstream file("test.txt");
    if (file.is_open())
    {
       string line;
        while (getline(file, line))
        {
           cout << line <<endl;
        }
        file.close();
    }
    else
```

```
    {
      cout << "Unable to open the file." <<endl;
    }
    return 0;
}
```

6.
```
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class FileCreateJava {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your name: ");
        String name = scanner.nextLine();
        System.out.print("Enter your roll number: ");
        int rollNumber = scanner.nextInt();

        try {
            FileWriter writer = new FileWriter("test1.txt");
            writer.write("Name: " + name + "\nRoll Number: " + rollNumber);
            writer.close();

            System.out.println("Data written to file successfully!");
        } catch (IOException e) {
```

```
            System.out.println("An error occurred while writing to the file.");
        }

        try (BufferedReader br = new BufferedReader(new FileReader("test1.txt"))) {
            System.out.println("Displaying the information :");
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
        }
        scanner.close();
    }
}
```

7.
```
#include <iostream>
#include <string>
using namespace std;
class Student
{
protected:
    int roll;
    string name;
    float mark;
public:
    void getInfo()
```

```
    {
      cout << "Enter Roll Number: ";
       cin >> roll;
       cin.ignore();
       cout << "Enter Name: ";
       getline(cin, name);
       cout << "Enter Mark: ";
       cin >> mark;
    }
    void displayInfo()
    {
       cout << "Roll Number: " << roll << endl;
       cout << "Name: " << name << endl;
       cout << "Mark: " << mark << endl;
    }
};
class Grade : public Student
{
private:
    string letterGrade;
public:
    void convertToLetterGrade()
    {
       if (mark >= 80)
          letterGrade = "A+";
       else if (mark >= 75 && mark < 80)
          letterGrade = "A";
```

```
        else
            letterGrade = "F";
    }
    void displayInfo()
    {
        Student::displayInfo();
        cout << "Letter Grade: " << letterGrade << endl;
    }
};

int main()
{
    Grade obj;
    obj.getInfo();
    obj.convertToLetterGrade();
    obj.displayInfo();
    return 0;
}
```

8.
```
#include <iostream>
using namespace std;
class Rectangle
{
private:
    double length;
```

```
    double width;
public:
    Rectangle(double l, double w)
    {
        length = l;
        width = w;
    }
    double calculateArea()
    {
        return length * width;
    }
};
class Square
{
private:
    double side;
public:
    Square(double s)
    {
        side = s;
    }
    double calculateArea()
    {
        return side * side;
    }
};
int main()
```

```
{
    double length = 20.0;
    double width = 6.0;
    double side = 6.0;

    Rectangle rectangle(length, width);
    Square square(side);

    double rectangleArea = rectangle.calculateArea();
    double squareArea = square.calculateArea();

    cout << "Area of Rectangle: " << rectangleArea << " square meters" << endl;
    cout << "Area of Square: " << squareArea << " square meters" << endl;
    return 0;
}
```

9.
```
public class Calculate {
    public static double calculate(double length, double width) {
        return length * width;
    }

    public static double calculate(double length, double width, double height) {
        return length * width * height;
    }

    public static void main(String[] args) {
        double length = 50.0;
        double width = 7.0;
```

```
        double height = 15.0;

        double area = calculate(length, width);
        double volume = calculate(length, width, height);

        System.out.println("Room Area: " + area + " square meters");
        System.out.println("Room Volume: " + volume + " cubic meters");
    }
}
```

10.
```
class NumberDisplayer {
    public void display(int num1) {
        System.out.println("Number 1: " + num1);
    }
}
class TwoNumberDisplayer extends NumberDisplayer {
    @Override
    public void display(int num1) {
        System.out.println("Overridden Number 1: " + num1);
    }
    public void display(int num1, int num2) {
        System.out.println("Number 1: " + num1);
        System.out.println("Number 2: " + num2);
    }
}
```

```
public class NumberDisplay {
    public static void main(String[] args) {
        int number1 = 10;
        int number2 = 20;
        NumberDisplayer displayer1 = new NumberDisplayer();
        displayer1.display(number1);
        TwoNumberDisplayer displayer2 = new TwoNumberDisplayer();
        displayer2.display(number1); // This will use the overridden method
        displayer2.display(number1, number2);
    }
}
```

11.
```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2;
    char operation;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;
    cout << "Enter operation (+, -, *, /): ";
    cin >> operation;
    switch (operation)
    {
    case '+':
```

```
        cout << "Addition: " << num1 + num2 << endl;
        break;
    case '-':
        cout << "Subtraction: " << num1 - num2 << endl;
        break;
    case '*':
        cout << "Multiplication: " << num1 * num2 << endl;
        break;
    case '/':
        if (num2 != 0)
        {
            cout << "Division: " << num1 / num2 << endl;
        }
        else
        {
            cout << "Cannot divide by zero!" << endl;
        }
        break;
    default:
        cout << "Invalid operation!" << endl;
    }
    return 0;
}
12. #include <iostream>
using namespace std;
class Product
{
```

```
private:
    string code;
    double price;
public:
    Product()
    {
        code = "";
        price = 0.0;
    }
    void setCode(string c)
    {
        code = c;
    }
    void setPrice(double p)
    {
        price = p;
    }
    string getCode()
    {
        return code;
    }
    string getPrice()
    {
        string formattedPrice = to_string(price);
        formattedPrice += " Tk";
        return formattedPrice;
    }
```

```
};
int main()
{
    int numProducts;
    cout << "Enter the number of products: ";
    cin >> numProducts;

    // Dynamically allocate an array of Product objects
    Product *products = new Product[numProducts];

    // Input product code and price
    for (int i = 0; i < numProducts; i++)
    {
        string code;
        double price;

        cout << "Enter code for product " << i + 1 << ": ";
        cin >> code;
        products[i].setCode(code);

        cout << "Enter price for product " << i + 1 << ": ";
        cin >> price;
        products[i].setPrice(price);
    }

    // Display product information using pointers
    for (int i = 0; i < numProducts; i++)
```

```
    {
        Product *p = &products[i];
        cout << "Product " << i + 1 << " code: " << p->getCode() << endl;
        cout << "Product " << i + 1 << " price: " << p->getPrice() << endl;
    }

    // Deallocate the dynamically allocated array
    delete[] products;

    return 0;
}
```