

JAVA PROGRAMMING: WORKSHOP – SOLUTIONS

SECTION 1

Exercise 1

Create a simple “Hello World” program.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Hello World

Exercise 2

Create a single-line comment before a line of code.

```
public class Main {  
    public static void main(String[] args) {  
        // This is a comment  
        System.out.println("Hello World");  
    }  
}
```

Exercise 3

Create a single-line comment at the end of a line of code.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World"); // This is a comment  
    }  
}
```

Exercise 4

Create a multi-level comment.

```
public class Main {  
    public static void main(String[] args) {  
        /* The code below will print the words Hello World  
        to the screen, and it is amazing */  
        System.out.println("Hello World");  
    }  
}
```

SECTION 2

Exercise 5

Create a string variable to display the name "John".

```
public class Main {
    public static void main(String[] args) {
        String name = "John";
        System.out.println(name);
    }
}
```

John

Exercise 6

Create an integer variable to display the number 15.

```
public class Main {
    public static void main(String[] args) {
        int myNum = 15;
        System.out.println(myNum);
    }
}
```

15

Exercise 7

Create a variable without assigning the value, and assign the value later.

```
public class Main {
    public static void main(String[] args) {
        int myNum;
        myNum = 15;
        System.out.println(myNum);
    }
}
```

15

Exercise 8

Overwrite an existing variable value.

```
public class Main {
    public static void main(String[] args) {
        int myNum = 15;
        myNum = 20; // myNum is now 20
        System.out.println(myNum);
    }
}
```

20

Exercise 9

Create a final variable (unchangeable and read-only).

```
public class Main {
    public static void main(String[] args) {
        final int myNum = 15;
        myNum = 20; // will generate an error
        System.out.println(myNum);
    }
}
```

```
Main.java:4: error: cannot assign a value to final variable myNum
    myNum = 20;
           ^
1 error
```

Exercise 10

Combine text and a variable on display, e.g. "Hello John".

```
public class Main {
    public static void main(String[] args) {
        String name = "John";
        System.out.println("Hello " + name);
    }
}
```

```
Hello John
```

Exercise 11

Add a variable to another variable, e.g. "John Travolta".

```
public class Main {
    public static void main(String[] args) {
        String firstName = "John ";
        String lastName = "Travolta";
        String fullName = firstName + lastName;
        System.out.println(fullName);
    }
}
```

```
John Travolta
```

Exercise 12

Declare many variables of the same type with a comma-separated list, and display the sum.
e.g. 5, 6, 50 → 61

```
public class Main {
    public static void main(String[] args) {
        int x = 5, y = 6, z = 50;
        System.out.println(x + y + z);
    }
}
```

```
61
```

Exercise 13

Create a demonstration of different data types in Java, e.g.

```
5
5.99
D
true
Hello
```

```
public class Main {
    public static void main(String[] args) {
        int myNum = 5;           // integer (whole number)
        float myFloatNum = 5.99f; // floating point number
        char myLetter = 'D';    // character
        boolean myBool = true;  // boolean
        String myText = "Hello"; // String
        System.out.println(myNum);
        System.out.println(myFloatNum);
        System.out.println(myLetter);
        System.out.println(myBool);
        System.out.println(myText);
    }
}
```

SECTION 3

Exercise 14

Assign a value of one primitive data type to another type.

(Widening Casting: converting a smaller type to a larger type size)

```
public class Main {
    public static void main(String[] args) {
        int myInt = 9;
        double myDouble = myInt; // Automatic casting: int to double

        System.out.println(myInt);
        System.out.println(myDouble);
    }
}
```

```
9
9.0
```

Exercise 15

Assign a value of one primitive data type to another type.
(Widening Casting: converting a larger type to a smaller size type)

```
public class Main {
    public static void main(String[] args) {
        double myDouble = 9.78d;
        int myInt = (int) myDouble; // Explicit casting: double to int

        System.out.println(myDouble);
        System.out.println(myInt);
    }
}
```

```
9.78
9
```

SECTION 4

Exercise 16

Create a program to add together two values.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x + y);
    }
}
```

Exercise 17

Create a program to subtract one value from another.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x - y);
    }
}
```

Exercise 18

Create a program to multiply two values.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x * y);
    }
}
```

Exercise 19

Create a program to divide one value by another.

```
public class Main {
    public static void main(String[] args) {
        int x = 12;
        int y = 3;
        System.out.println(x / y);
    }
}
```

Exercise 20

Create a program to return the division.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 2;
        System.out.println(x % y);
    }
}
```

Exercise 21

Create a program to increase the value of a variable by 1.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        ++x;
        System.out.println(x);
    }
}
```

Exercise 22

Create a program to decreases the value of a variable by 1.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        --x;
        System.out.println(x);
    }
}
```

Exercise 23

Create a program that uses an assignment operator to assign the value 10 to a variable called x.

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        System.out.println(x);
    }
}
```

Exercise 24

Create a program that uses an assignment operator to add a value to a variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        x += 3;
        System.out.println(x);
    }
}
```

Exercise 25

Create a program that uses an assignment operator to subtract a value from a variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        x -= 3;
        System.out.println(x);
    }
}
```

Exercise 26

Create a program that uses an assignment operator to multiply a value with a variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        x *= 3;
        System.out.println(x);
    }
}
```

Exercise 27

Create a program that uses an assignment operator to divide a value by another variable.

```
public class Main {
    public static void main(String[] args) {
        double x = 5;
        x /= 3;
        System.out.println(x);
    }
}
```

Exercise 28

Create a program that uses an assignment operator to calculate the remainder of a division operation between the value and a variable.

```
public class Main {
    public static void main(String[] args) {
        double x = 5;
        x %= 3;
        System.out.println(x);
    }
}
```

Exercise 29

Create a program that uses an assignment operator to perform a bitwise AND operation between a value and a variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 12;
        x &= 6;
        System.out.println(x);
    }
}
```

```
int x = 12; // Binary: 1100
int y = 6; // Binary: 0110
x &= y; // Equivalent to: x = x & y;
System.out.println(x); // Output: 4 (Binary: 0100)
```

Exercise 30

Create a program that uses an assignment operator to perform a bitwise OR operation between a value and a variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 12;
        x |= 6;
        System.out.println(x);
    }
}
```

```
int x = 12; // Binary: 1100
int y = 6; // Binary: 0110
x |= y; // Equivalent to: x = x | y;
System.out.println(x); // Output: 14 (Binary: 1110)
```

Exercise 31

Create a program that uses an assignment operator to perform a bitwise XOR operation between a value and a variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 12;
        x ^= 6;
        System.out.println(x);
    }
}
```

```
int x = 12; // Binary: 1100
int y = 6; // Binary: 0110
x ^= y; // Equivalent to: x = x ^ y;
System.out.println(x); // Output: 10 (Binary: 1010)
```


Exercise 32

Create a program that uses an assignment operator to perform a signed right shift operation on a value, using the number of positions specified.

```
public class Main {
    public static void main(String[] args) {
        int x = 32;
        x >>= 2;
        System.out.println(x);
    }
}
```

```
int x = 32; // Binary: 100000
int shiftAmount = 2;
x >>= shiftAmount; // Equivalent to: x = x >> shiftAmount;
System.out.println(x); // Output: 8 (Binary: 1000)
```

Exercise 33

Create a program that uses an assignment operator to perform a signed left shift operation on a value, using the number of positions specified.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        x <<= 2;
        System.out.println(x);
    }
}
```

```
int x = 5; // Binary: 00000101
int shiftAmount = 2;
x <<= shiftAmount; // Equivalent to: x = x << shiftAmount;
System.out.println(x); // Output: 20 (Binary: 00010100)
```

Exercise 34

Create a program that uses an comparison operator to find out if a variable is equal to another variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x == y); // returns false because 5 is not equal to 3
    }
}
```

Exercise 35

Create a program that uses an comparison operator to find out if a variable is not equal to another variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x != y); // returns true because 5 is not equal to 3
    }
}
```

Exercise 36

Create a program that uses an comparison operator to find out if a variable is greater than another variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x > y); // returns true because 5 is greater than 3
    }
}
```

Exercise 37

Create a program that uses an comparison operator to find out if a variable is less than another variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x < y); // returns true because 5 is less than 3
    }
}
```

Exercise 38

Create a program that uses an comparison operator to find out if a variable is greater than or equal to another variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x >= y); // returns true because 5 is greater, or equal, to 3
    }
}
```

Exercise 39

Create a program that uses an comparison operator to find out if a variable is less than or equal to another variable.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        System.out.println(x <= y); // returns true because 5 is less, or equal, to 3
    }
}
```

Exercise 40

Create a program that uses an logical operator to find out if two statements are true.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        System.out.println(x > 3 && x < 10); // returns true because 5 is greater than 3 AND 5 is
        less than 10
    }
}
```

True

Exercise 41

Create a program that uses an logical operator to find out if one of statements is true.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        System.out.println(x > 3 || x < 4); // returns true because one of the conditions are true (5
        is greater than 3, but 5 is not less than 4)
    }
}
```

True

Exercise 42

Create a program that uses an logical operator to reverse the result.
e.g. returns false if the result is true.

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        System.out.println(!(x > 3 && x < 10)); // returns false because ! (not) is used to reverse
        the result
    }
}
```

False

SECTION 5

Exercise 43

Create a variable of type String and assign it a value.

```
public class Main {
    public static void main(String[] args) {
        String greeting = "Hello";
        System.out.println(greeting);
    }
}
```

Exercise 44

Create two string variables to display the text “It's alright” and “That's great” in separate lines.

```
public class Main {
    public static void main(String[] args) {
        String txt1 = "It's alright";
        String txt2 = "That's great";
        System.out.println(txt1);
        System.out.println(txt2);
    }
}
```

```
It's alright
That's great
```

Exercise 45

Create a program to find the length of the string “ABCDEFGHIJKLMNOPQRSTUVWXYZ”.

```
public class Main {
    public static void main(String[] args) {
        String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        System.out.println("The length of the txt string is: " + txt.length());
    }
}
```

```
The length of the txt string is: 26
```

Exercise 46

Create a program to convert strings to uppercase and lowercase.

```
public class Main {
    public static void main(String[] args) {
        String txt = "Hello World";
        System.out.println(txt.toUpperCase());
        System.out.println(txt.toLowerCase());
    }
}
```

```
HELLO WORLD
hello world
```

Exercise 47

Create a program to find a string in a string.

```
public class Main {
    public static void main(String[] args) {
        String txt = "Please locate where 'locate' occurs!";
        System.out.println(txt.indexOf("locate"));
    }
}
```

```
7
```

Exercise 48

Create a program to concatenate two strings in one line.

```
public class Main {
    public static void main(String args[]) {
        String firstName = "John";
        String lastName = "Taylor";
        System.out.println(firstName + " " + lastName);
    }
}
```

John Taylor

Exercise 49

Create a program to concatenate a string variable and an integer variable in one line.

```
public class Main {
    public static void main(String[] args) {
        String x = "10";
        int y = 20;
        String z = x + y;
        System.out.println(z);
    }
}
```

1020

SECTION 6

Exercise 50

Create a program to return the highest value of x and y.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(Math.max(5, 10));
    }
}
```

Exercise 51

Create a program to return the lowest value of x and y.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(Math.min(5, 10));
    }
}
```

Exercise 52

Create a program to return the square root of x.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(Math.sqrt(64));
    }
}
```

Exercise 53

Create a program to return the absolute (positive) value of x.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(Math.abs(-4.7));
    }
}
```

4.7

Exercise 54

Create a program to return a random number between 0 and 1.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(Math.random());
    }
}
```

0.0020118502459267606

SECTION 7

Exercise 55

Create two Boolean types in a program.

```
public class Main {
    public static void main(String[] args) {
        boolean isJavaFun = true;
        boolean isFishTasty = false;
        System.out.println(isJavaFun);
        System.out.println(isFishTasty);
    }
}
```

true
false

Exercise 56

Create a program to find out if an expression is true or false.

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 9;
        System.out.println(x > y); // returns true, because 10 is higher than 9
    }
}
```

Exercise 57

Create a program to use the “equal to” operator to evaluate a Boolean expression.

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        System.out.println(x == 10); // returns true, because the value of x is equal to 10
    }
}
```

SECTION 8

Exercise 58

Create a program to use the if statement to specify a block of Java code to be executed if a condition is true.

```
public class Main {
    public static void main(String[] args) {
        if (20 > 18) {
            System.out.println("20 is greater than 18"); // obviously
        }
    }
}
```

20 is greater than 18

Exercise 59

Create a program to use the else statement to specify a block of code to be executed if the condition is false.

```
public class Main {
    public static void main(String[] args) {
        int time = 20;
        if (time < 18) {
            System.out.println("Good day.");
        } else {
            System.out.println("Good evening.");
        }
    }
}
```

Good evening

Exercise 60

Create a program to use the else if statement to specify a new condition if the first condition is false.

```
public class Main {
    public static void main(String[] args) {
        int time = 22;
        if (time < 10) {
            System.out.println("Good morning.");
        } else if (time < 18) {
            System.out.println("Good day.");
        } else {
            System.out.println("Good evening.");
        }
    }
}
```

Good evening

SECTION 9

Exercise 61

Create a program to use the switch statement, to use the weekday number to calculate the weekday name.

```
public class Main {
    public static void main(String[] args) {
        int day = 4;
        switch (day) {
            case 1:
                System.out.println("Monday");
                break;
            case 2:
                System.out.println("Tuesday");
                break;
            case 3:
                System.out.println("Wednesday");
                break;
            case 4:
                System.out.println("Thursday");
                break;
            case 5:
                System.out.println("Friday");
                break;
            case 6:
                System.out.println("Saturday");
                break;
            case 7:
                System.out.println("Sunday");
                break;
        }
    }
}
```

Thursday

Exercise 62

Create a program to use the switch statement with a default keyword.

```
public class Main {
    public static void main(String[] args) {
        int day = 4;
        switch (day) {
            case 6:
                System.out.println("Today is Saturday");
                break;
            case 7:
                System.out.println("Today is Sunday");
                break;
            default:
                System.out.println("Looking forward to the Weekend");
        }
    }
}
```

Looking forward to the Weekend

SECTION 10

Exercise 63

Create a program to use While loop to display numbers 0-4 in different lines.

```
public class Main {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            i++;
        }
    }
}
```

0
1
2
3
4

Exercise 64

Create a program to use Do While loop to display numbers 0-4 in different lines.

```
public class Main {
    public static void main(String[] args) {
        int i = 0;
        do {
            System.out.println(i);
            i++;
        }
        while (i < 5);
    }
}
```

0
1
2
3
4

Exercise 65

Create a program to use For loop to display numbers 0-4 in different lines.

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            System.out.println(i);
        }
    }
}
```

```
0
1
2
3
4
```

Exercise 66

Create a program to use For-Each loop to display the names of 4 cars.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        for (String i : cars) {
            System.out.println(i);
        }
    }
}
```

```
Volvo
BMW
Ford
Mazda
```

Exercise 67

Create a program to Break a For loop to display numbers 0-3 in different lines.

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) {
            if (i == 4) {
                break;
            }
            System.out.println(i);
        }
    }
}
```

```
0
1
2
3
```

Exercise 68

Create a program to Continue a For loop to display numbers 0-9 in different lines.

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) {
            if (i == 4) {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

```
0
1
2
3
5
6
7
8
9
```

SECTION 11

Exercise 69

Create a program to create and access an array.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        System.out.println(cars[0]);
    }
}
```

Volvo

Exercise 70

Create a program to change an array element.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        cars[0] = "Opel";
        System.out.println(cars[0]);
    }
}
```

Opel

Exercise 71

Create a program to find the length of an array.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        System.out.println(cars.length);
    }
}
```

Exercise 72

Create a program to access an array.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        cars[0] = "Opel";
        System.out.println(cars[0]);
    }
}
```

Exercise 73

Create a program to loop through an array.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        for (int i = 0; i < cars.length; i++) {
            System.out.println(cars[i]);
        }
    }
}
```

Exercise 74

Create a program to loop through an array with for-each.

```
public class Main {
    public static void main(String[] args) {
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
        for (String i : cars) {
            System.out.println(i);
        }
    }
}
```

Exercise 75

Create a program to create and access a multidimensional array.

```
public class Main {
    public static void main(String[] args) {
        int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
        System.out.println(myNumbers[1][2]);
    }
}
```

Exercise 76

Create a program to loop through a multidimensional array.

```
public class Main {
    public static void main(String[] args) {
        int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
        for (int i = 0; i < myNumbers.length; ++i) {
            for(int j = 0; j < myNumbers[i].length; ++j) {
                System.out.println(myNumbers[i][j]);
            }
        }
    }
}
```

SECTION 12

Exercise 77

Create a program to create and call a method.

```
public class Main {
    static void myMethod() {
        System.out.println("I just got executed!");
    }
    public static void main(String[] args) {
        myMethod();
    }
}
```

Exercise 78

Create a program to call a method multiple times.

```
public class Main {
    static void myMethod() {
        System.out.println("I just got executed!");
    }
    public static void main(String[] args) {
        myMethod();
        myMethod();
        myMethod();
    }
}
```

Exercise 79

Create a program to use a method with multiple parameters.

```
public class Main {
    static void myMethod(String fname, int age) {
        System.out.println(fname + " is " + age);
    }
    public static void main(String[] args) {
        myMethod("Liam", 5);
        myMethod("Jenny", 8);
        myMethod("Anja", 31);
    }
}
```

```
Liam is 5
Jenny is 8
Anja is 31
```

Exercise 80

Create a program to return a value.

```
public class Main {
    static int myMethod(int x) {
        return 5 + x;
    }
    public static void main(String[] args) {
        System.out.println(myMethod(3));
    }
}
```

Exercise 81

Create a program to use a method with if... else.

```
public class Main {

    // Create a checkAge() method with an integer parameter called age
    static void checkAge(int age) {

        // If age is less than 18, print "access denied"
        if (age < 18) {
            System.out.println("Access denied - You are not old enough!");
        }

        // If age is greater than, or equal to, 18, print "access granted"
        } else {
            System.out.println("Access granted - You are old enough!");
        }
    }

    public static void main(String[] args) {
        checkAge(20); // Call the checkAge method and pass along an age of 20
    }
}
```

Exercise 82

Create a program to use a method loading.

```
public class Main {
    static int plusMethod(int x, int y) {
        return x + y;
    }
    static double plusMethod(double x, double y) {
        return x + y;
    }
    public static void main(String[] args) {
        int myNum1 = plusMethod(8, 5);
        double myNum2 = plusMethod(4.3, 6.26);
        System.out.println("int: " + myNum1);
        System.out.println("double: " + myNum2);
    }
}
```

```
int: 13
double: 10.559999999999999
```

Exercise 83

Create a program to use recursion.

```
public class Main {
    public static void main(String[] args) {
        int result = sum(10);
        System.out.println(result);
    }
    public static int sum(int k) {
        if (k > 0) {
            return k + sum(k - 1);
        } else {
            return 0;
        }
    }
}
```

Exercise 84

Create a program to use recursion function with a halting condition.

```
public class Main {
    public static void main(String[] args) {
        int result = sum(5, 10);
        System.out.println(result);
    }
    public static int sum(int start, int end) {
        if (end > start) {
            return end + sum(start, end - 1);
        } else {
            return end;
        }
    }
}
```

SECTION 13

Exercise 85

Create a class and an object of a class.

```
public class Main {
    int x = 5;
    public static void main(String[] args) {
        Main myObj = new Main();
        System.out.println(myObj.x);
    }
}
```

5

Exercise 86

Create multiple objects of a class.

```
public class Main {
    int x = 5;
    public static void main(String[] args) {
        Main myObj1 = new Main();
        Main myObj2 = new Main();
        System.out.println(myObj1.x);
        System.out.println(myObj2.x);
    }
}
```

5
5

Exercise 87

Create a program to access class attributes (variables).

```
public class Main {
    int x = 5;
    public static void main(String[] args) {
        Main myObj = new Main();
        System.out.println(myObj.x);
    }
}
```

5

Exercise 88

Create a program to modify attributes.

```
public class Main {
    int x;
    public static void main(String[] args) {
        Main myObj = new Main();
        myObj.x = 40;
        System.out.println(myObj.x);
    }
}
```

40

Exercise 89

Create a program to override existing attributes values.

```
public class Main {
    int x = 10;
    public static void main(String[] args) {
        Main myObj = new Main();
        myObj.x = 25; // x is now 25
        System.out.println(myObj.x);
    }
}
```

25

Exercise 90

Create a program to use multiple attributes.

```
public class Main {
    String fname = "John";
    String lname = "Doe";
    int age = 24;
    public static void main(String[] args) {
        Main myObj = new Main();
        System.out.println("Name: " + myObj.fname + " " + myObj.lname);
        System.out.println("Age: " + myObj.age);
    }
}
```

```
Name: John Doe
Age: 24
```

Exercise 91

Create a program to create a class method.

```
public class Main {
    static void myMethod() {
        System.out.println("Hello World!");
    }
    public static void main(String[] args) {
        myMethod();
    }
}
```

```
Hello World!
```

Exercise 92

Create a program to use a method with parameters.

```
public class Main {
    static void myMethod(int x) {
        System.out.println(x);
    }
    public static void main(String[] args) {
        myMethod(10);
    }
}
```

10

Exercise 93

Create a program to access class methods with an object.

```
// Create a Main class
public class Main {

    // Create a fullThrottle() method
    public void fullThrottle() {
        System.out.println("The car is going as fast as it can!");
    }

    // Create a speed() method and add a parameter
    public void speed(int maxSpeed) {
        System.out.println("Max speed is: " + maxSpeed);
    }

    // Inside main, call the methods on the myCar object
    public static void main(String[] args) {
        Main myCar = new Main(); // Create a myCar object
        myCar.fullThrottle(); // Call the fullThrottle() method
        myCar.speed(200); // Call the speed() method
    }
}
```

```
The car is going as fast as it can!
Max speed is: 200
```

Exercise 94

Create a program to create a class constructor.

```
// Create a Main class
public class Main {
    int x;

    // Create a class constructor for the Main class
    public Main() {
        x = 5;
    }

    public static void main(String[] args) {
        Main myObj = new Main();
        System.out.println(myObj.x);
    }
}
```

5

Exercise 95

Create a program to use constructor with parameters.

```
//filename: Main.java
public class Main {
    int modelYear;
    String modelName;

    public Main(int year, String name) {
        modelYear = year;
        modelName = name;
    }

    public static void main(String[] args) {
        Main myCar = new Main(1969, "Mustang");
        System.out.println(myCar.modelYear + " " + myCar.modelName);
    }
}
```

1969 Mustang

SECTION 14

Exercise 96

Create a program with public class.

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

```
Hello World
```

Exercise 97

Create a program with default class.

```
class MyClass {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

```
Hello World
```

Exercise 98

Create a program with public attributes.

```
class MyClass {
    public static void main(String[] args) {
        Person myObj = new Person();
        System.out.println("Name: " + myObj.fname + " " + myObj.lname);
        System.out.println("Email: " + myObj.email);
        System.out.println("Age: " + myObj.age);
    }
}
```

```
Name: John Doe
Email: john@doe.com
Age: 24
```

Exercise 99

Create a program with private attributes.

```
public class Main {
    private String fname = "John";
    private String lname = "Doe";
    private String email = "john@doe.com";
    private int age = 24;
    public static void main(String[] args) {
        Main myObj = new Main();
        System.out.println("Name: " + myObj.fname + " " + myObj.lname);
        System.out.println("Email: " + myObj.email);
        System.out.println("Age: " + myObj.age);
    }
}
```

```
Name: John Doe
Email: john@doe.com
Age: 24
```

Exercise 100

Create a program with default attributes.

```
class Person {
    String fname = "John";
    String lname = "Doe";
    String email = "john@doe.com";
    int age = 24;

    public static void main(String[] args) {
        Person myObj = new Person();
        System.out.println("Name: " + myObj.fname + " " + myObj.lname);
        System.out.println("Email: " + myObj.email);
        System.out.println("Age: " + myObj.age);
    }
}
```

```
Name: John Doe
Email: john@doe.com
Age: 24
```

Exercise 101

Create a program with protected attributes.

```
class Person {
    protected String fname = "John";
    protected String lname = "Doe";
    protected String email = "john@doe.com";
    protected int age = 24;
}

class Student extends Person {
    private int graduationYear = 2018;
    public static void main(String[] args) {
        Student myObj = new Student();
        System.out.println("Name: " + myObj.fname + " " + myObj.lname);
        System.out.println("Email: " + myObj.email);
        System.out.println("Age: " + myObj.age);
        System.out.println("Graduation Year: " + myObj.graduationYear);
    }
}
```

```
Name: John Doe
Email: john@doe.com
Age: 24
Graduation Year: 2018
```

Exercise 102

Create an abstract class with abstract methods.

```
class MyClass {
    public static void main(String[] args) {
        // create an object of the Student class (which inherits attributes and methods from
        Person)
        Student myObj = new Student();

        System.out.println("Name: " + myObj.fname + " " + myObj.lname);
        System.out.println("Email: " + myObj.email);
    }
}
```

MyClass.java

```

System.out.println("Age: " + myObj.age);
System.out.println("Graduation Year: " + myObj.graduationYear);
myObj.study(); // call abstract method
}
}

```

Person.java

```

// abstract class
abstract class Person {
    public String fname = "John";
    public String lname = "Doe";
    public String email = "john@doe.com";
    public int age = 24;
    public abstract void study(); // abstract method
}

// Subclass (inherit from Person)
class Student extends Person {
    public int graduationYear = 2018;
    public void study() {
        System.out.println("Studying all day long");
    }
}

```

```

Name: John Doe
Email: john@doe.com
Age: 24
Graduation Year: 2018
Studying all day long

```

SECTION 15

Exercise 103

Create a program to use encapsulation (getters and setters)

```
public class Main {
    public static void main(String[] args) {
        Person myObj = new Person();
        myObj.setName("John");
        System.out.println(myObj.getName());
    }
}
```

MyClass.java

```
public class Person {
    private String name;

    // Getter
    public String getName() {
        return name;
    }

    // Setter
    public void setName(String newName) {
        this.name = newName;
    }
}
```

Person.java

John

Exercise 104

Create a program to import a class from the Java API.

```
import java.util.Scanner; // import the Scanner class

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String userName;

        // Enter username and press Enter
        System.out.println("Enter username");
        userName = myObj.nextLine();

        System.out.println("Username is: " + userName);
    }
}
```

```
Enter username
DANIEL
Username is: DANIEL
```

Exercise 105

Create a program to import a package from the Java API.

```
import java.util.*; // import the java.util package

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String userName;

        // Enter username and press EnterS
        System.out.println("Enter username");
        userName = myObj.nextLine();

        System.out.println("Username is: " + userName);
    }
}
```

```
Enter username
DANIEL
Username is: DANIEL
```

Exercise 106

Create a package.

```
package mypack;

class MyPackageClass {
    public static void main(String[] args) {
        System.out.println("This is my package!");
    }
}
```

```
This is my package!
```

Exercise 107

Create a program to use Java inheritance (extends).

```
class Vehicle {
    protected String brand = "Ford";
    public void honk() {
        System.out.println("Tuut, tuut!");
    }
}

class Car extends Vehicle {
    private String modelName = "Mustang";
    public static void main(String[] args) {
        Car myFastCar = new Car();
        myFastCar.honk();
        System.out.println(myFastCar.brand + " " + myFastCar.modelName);
    }
}
```

```
Tuut, tuut!
Ford Mustang
```


Exercise 108

Create a program to use Java polymorphism.

```

class Animal {
    public void animalSound() {
        System.out.println("The animal makes a sound");
    }
}

class Pig extends Animal {
    public void animalSound() {
        System.out.println("The pig says: wee wee");
    }
}

class Dog extends Animal {
    public void animalSound() {
        System.out.println("The dog says: bow wow");
    }
}

class Main {
    public static void main(String[] args) {
        Animal myAnimal = new Animal();
        Animal myPig = new Pig();
        Animal myDog = new Dog();

        myAnimal.animalSound();
        myPig.animalSound();
        myDog.animalSound();
    }
}

```

```

The animal makes a sound
The pig says: wee wee
The dog says: bow wow

```

SECTION 16**Exercise 109**

Create a program to use inner class.

```
class OuterClass {
    int x = 10;

    class InnerClass {
        int y = 5;
    }
}
public class Main {
    public static void main(String[] args) {
        OuterClass myOuter = new OuterClass();
        OuterClass.InnerClass myInner = myOuter.new InnerClass();
        System.out.println(myInner.y + myOuter.x);
    }
}
```

15**Exercise 110**

Create a program to use private inner class (error when trying to access it from an outside class).

```
class OuterClass {
    int x = 10;

    private class InnerClass {
        int y = 5;
    }
}
public class Main {
    public static void main(String[] args) {
        OuterClass myOuter = new OuterClass();
        OuterClass.InnerClass myInner = myOuter.new InnerClass();
        System.out.println(myInner.y + myOuter.x);
    }
}
```

```
Main.java:13: error: OuterClass.InnerClass has private access in OuterClass
    OuterClass.InnerClass myInner = myOuter.new InnerClass();
            ^
Main.java:13: error: OuterClass.InnerClass has private access in OuterClass
    OuterClass.InnerClass myInner = myOuter.new InnerClass();
            ^
2 errors
```

Exercise 111

Create a program to use static inner class.

```
class OuterClass {
    int x = 10;

    static class InnerClass {
        int y = 5;
    }
}

public class Main {
    public static void main(String[] args) {
        OuterClass.InnerClass myInner = new OuterClass.InnerClass();
        System.out.println(myInner.y);
    }
}
```

5

Exercise 112

Create a program to access outer class from inner class.

```
class OuterClass {
    int x = 10;

    class InnerClass {
        public int myInnerMethod() {
            return x;
        }
    }
}

public class Main {
    public static void main(String[] args) {
        OuterClass myOuter = new OuterClass();
        OuterClass.InnerClass myInner = myOuter.new InnerClass();
        System.out.println(myInner.myInnerMethod());
    }
}
```

10

Exercise 113

Create a program to use Java abstract classes and methods.

```
// Abstract class
abstract class Animal {
    // Abstract method (does not have a body)
    public abstract void animalSound();
    // Regular method
    public void sleep() {
        System.out.println("Zzz");
    }
}
```

```

}
}

// Subclass (inherit from Animal)
class Pig extends Animal {
    public void animalSound() {
        // The body of animalSound() is provided here
        System.out.println("The pig says: wee wee");
    }
}

class Main {
    public static void main(String[] args) {
        Pig myPig = new Pig(); // Create a Pig object
        myPig.animalSound();
        myPig.sleep();
    }
}

```

```

The pig says: wee wee
Zzz

```

Exercise 114

Create a program to use Java interface (implements)

```

interface Animal {
    public void animalSound(); // interface method (does not have a body)
    public void sleep(); // interface method (does not have a body)
}

class Pig implements Animal {
    public void animalSound() {
        System.out.println("The pig says: wee wee");
    }
    public void sleep() {
        System.out.println("Zzz");
    }
}

class Main {
    public static void main(String[] args) {
        Pig myPig = new Pig();
        myPig.animalSound();
        myPig.sleep();
    }
}

```

```

The pig says: wee wee
Zzz

```

Exercise 115

Create a program to use multiple interfaces.

```
interface FirstInterface {
    public void myMethod(); // interface method
}

interface SecondInterface {
    public void myOtherMethod(); // interface method
}

// DemoClass "implements" FirstInterface and SecondInterface
class DemoClass implements FirstInterface, SecondInterface {
    public void myMethod() {
        System.out.println("Some text..");
    }
    public void myOtherMethod() {
        System.out.println("Some other text...");
    }
}

class Main {
    public static void main(String[] args) {
        DemoClass myObj = new DemoClass();
        myObj.myMethod();
        myObj.myOtherMethod();
    }
}
```

```
Some text...
```

```
Some other text...
```

SECTION 17

Exercise 116

Create an enum with a group of constants.

```
enum Level {  
    LOW,  
    MEDIUM,  
    HIGH  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Level myVar = Level.MEDIUM;  
        System.out.println(myVar);  
    }  
}
```

MEDIUM

Exercise 117

Create a program to use an enum inside a class.

```
public class Main {  
    enum Level {  
        LOW,  
        MEDIUM,  
        HIGH  
    }  
  
    public static void main(String[] args) {  
        Level myVar = Level.MEDIUM;  
        System.out.println(myVar);  
    }  
}
```

MEDIUM

Exercise 118

Create a program to use an enum in a switch statement.

```
enum Level {  
    LOW,  
    MEDIUM,  
    HIGH  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Level myVar = Level.MEDIUM;  
  
        switch(myVar) {  
            case LOW:
```

```

        System.out.println("Low level");
        break;
    case MEDIUM:
        System.out.println("Medium level");
        break;
    case HIGH:
        System.out.println("High level");
        break;
    }
}
}

```

Medium level

Exercise 119

Create a program to use loop through an enum.

```

enum Level {
    LOW,
    MEDIUM,
    HIGH
}

public class Main {
    public static void main(String[] args) {
        for (Level myVar : Level.values()) {
            System.out.println(myVar);
        }
    }
}

```

LOW
MEDIUM
HIGH

Exercise 120

Create a program to read user input.

```

import java.util.Scanner; // import the Scanner class

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String userName;

        // Enter username and press Enter
        System.out.println("Enter username");
        userName = myObj.nextLine();

        System.out.println("Username is: " + userName);
    }
}

```

Enter username
DANIEL
Username is: DANIEL

Exercise 121

Create a program to read user input of various types.

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);

        System.out.println("Enter name, age and salary:");

        // String input
        String name = myObj.nextLine();

        // Numerical input
        int age = myObj.nextInt();
        double salary = myObj.nextDouble();

        // Output input by user
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Salary: " + salary);
    }
}
```

```
Enter name, age and salary:
DANIEL
25
3000
Name: DANIEL
Age: 25
Salary: 3000
```

Exercise 122

Create a program to display current date.

```
import java.time.LocalDate; // import the LocalDate class
public class Main {
    public static void main(String[] args) {
        LocalDate myObj = LocalDate.now(); // Create a date object
        System.out.println(myObj); // Display the current date
    }
}
```

2023-08-04

Exercise 123

Create a program to display current time.

```
import java.time.LocalTime; // import the LocalTime class
public class Main {
    public static void main(String[] args) {
        LocalTime myObj = LocalTime.now();
        System.out.println(myObj);
    }
}
```

16:51:02.961559

Exercise 124

Create a program to display current date and time.

```
import java.time.LocalDateTime; // import the LocalDateTime class
public class Main {
    public static void main(String[] args) {
        LocalDateTime myObj = LocalDateTime.now();
        System.out.println(myObj);
    }
}
```

```
2023-08-04T16:51:48.985372
```

Exercise 125

Create a program to format date and time.

```
import java.time.LocalDateTime; // Import the LocalDateTime class
import java.time.format.DateTimeFormatter; // Import the DateTimeFormatter class
public class Main {
    public static void main(String[] args) {
        LocalDateTime myDateObj = LocalDateTime.now();
        System.out.println("Before formatting: " + myDateObj);
        DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yyyy
HH:mm:ss");

        String formattedDate = myDateObj.format(myFormatObj);
        System.out.println("After formatting: " + formattedDate);
    }
}
```

```
Before Formatting: 2023-08-04T16:52:30.737316
After Formatting: 04-08-2023 16:52:30
```

SECTION 18

Exercise 126

Create an ArrayList.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        System.out.println(cars);
    }
}
```

```
[Volvo, BMW, Ford, Mazda]
```

Exercise 127

Create a program to access an item in an ArrayList.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        System.out.println(cars.get(0));
    }
}
```

```
Volvo
```

Exercise 128

Create a program to remove an item from an ArrayList.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        cars.remove(0);
        System.out.println(cars);
    }
}
```

```
[BMW, Ford, Mazda]
```

Exercise 129

Create a program to remove all items from an ArrayList.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        cars.clear();
        System.out.println(cars);
    }
}
```



Exercise 130

Create a program to get the size of an ArrayList.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        System.out.println(cars.size());
    }
}
```



Exercise 131

Create a program to loop through an ArrayList.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        for (int i = 0; i < cars.size(); i++) {
            System.out.println(cars.get(i));
        }
    }
}
```

```
Volvo
BMW
Ford
Mazda
```

Exercise 132

Create a program to loop through an ArrayList with for-each.

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        for (String i : cars) {
            System.out.println(i);
        }
    }
}
```

```
Volvo
BMW
Ford
Mazda
```

Exercise 133

Create an ArrayList that should store numbers (integers)

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> myNumbers = new ArrayList<Integer>();
        myNumbers.add(10);
        myNumbers.add(15);
        myNumbers.add(20);
        myNumbers.add(25);
        for (int i : myNumbers) {
            System.out.println(i);
        }
    }
}
```

```
10
15
20
25
```

SECTION 19**Exercise 134**

Create a HashMap.

```
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        HashMap<String, String> capitalCities = new HashMap<String, String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
        System.out.println(capitalCities);
    }
}
```

```
{USA=Washington DC, Norway=Oslo, England=London, Germany=Berlin}
```

Exercise 135

Access an item in a HashMap.

```
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        HashMap<String, String> capitalCities = new HashMap<String, String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
        System.out.println(capitalCities.get("England"));
    }
}
```

```
London
```

Exercise 136

Remove an item from a HashMap.

```
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        HashMap<String, String> capitalCities = new HashMap<String, String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
        capitalCities.remove("England");
        System.out.println(capitalCities);
    }
}
```

```
{USA=Washington DC, Norway=Oslo, Germany=Berlin}
```

Exercise 137

Access all items from a HashMap.

```
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        HashMap<String, String> capitalCities = new HashMap<String, String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
        capitalCities.clear();
        System.out.println(capitalCities);
    }
}
```



Exercise 138

Get the size of a HashMap.

```
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        HashMap<String, String> capitalCities = new HashMap<String, String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
        System.out.println(capitalCities.size());
    }
}
```



Exercise 139

Loop through a HashMap.

```
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        HashMap<String, String> capitalCities = new HashMap<String, String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");

        for (String i : capitalCities.keySet()) {
            System.out.println("key: " + i + " value: " + capitalCities.get(i));
        }
    }
}
```

```
key: USA value: Washington DC
key: Norway value: Oslo
key: England value: London
key: Germany value: Berlin
```

Exercise 140

Create a HashMap that should store String keys and Integer values.

```
// Import the HashMap class
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {

        // Create a HashMap object called people
        HashMap<String, Integer> people = new HashMap<String, Integer>();

        // Add keys and values (Name, Age)
        people.put("John", 32);
        people.put("Steve", 30);
        people.put("Angie", 33);

        for (String i : people.keySet()) {
            System.out.println("Name: " + i + " Age: " + people.get(i));
        }
    }
}
```

```
Name: Angie Age: 33
Name: Steve Age: 30
Name: John Age: 32
```

SECTION 20

Exercise 141

Create a HashSet and add items to it.

```
// Import the HashSet class
import java.util.HashSet;
public class Main {
    public static void main(String[] args) {
        HashSet<String> cars = new HashSet<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("BMW");
        cars.add("Mazda");
        System.out.println(cars);
    }
}
```

[Volvo, Mazda, Ford, BMW]

Exercise 142

Check if an item exists in a HashSet.

```
// Import the HashSet class
import java.util.HashSet;
public class Main {
    public static void main(String[] args) {
        HashSet<String> cars = new HashSet<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("BMW");
        cars.add("Mazda");
        System.out.println(cars.contains("Mazda"));
    }
}
```

True

Exercise 143

Remove an item from a HashSet.

```
// Import the HashSet class
import java.util.HashSet;
public class Main {
    public static void main(String[] args) {
        HashSet<String> cars = new HashSet<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("BMW");
        cars.add("Mazda");
        cars.remove("Volvo");
        System.out.println(cars);
    }
}
```

[Mazda, Ford, BMW]

Exercise 144

Remove all items from a HashSet.

```
// Import the HashSet class
import java.util.HashSet;
public class Main {
    public static void main(String[] args) {
        HashSet<String> cars = new HashSet<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("BMW");
        cars.add("Mazda");
        cars.clear();
        System.out.println(cars);
    }
}
```

[]

Exercise 145

Get the size of a HashSet.

```
// Import the HashSet class
import java.util.HashSet;
public class Main {
    public static void main(String[] args) {
        HashSet<String> cars = new HashSet<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("BMW");
        cars.add("Mazda");
        System.out.println(cars.size());
    }
}
```

4

Exercise 146

Loop through a HashSet.

```
// Import the HashSet class
import java.util.HashSet;
public class Main {
    public static void main(String[] args) {
        HashSet<String> cars = new HashSet<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("BMW");
        cars.add("Mazda");
        for (String i : cars) {
            System.out.println(i);
        }
    }
}
```

```
Volvo
Mazda
Ford
BMW
```

Exercise 147

Create a HashSet that should store Integer values.

```
// Import the HashSet class
import java.util.HashSet;

public class Main {
    public static void main(String[] args) {

        // Create a HashSet object called numbers
        HashSet<Integer> numbers = new HashSet<Integer>();

        // Add values to the set
        numbers.add(4);
        numbers.add(7);
        numbers.add(8);

        // Show which numbers between 1 and 10 are in the set
        for(int i = 1; i <= 10; i++) {
            if(numbers.contains(i)) {
                System.out.println(i + " was found in the set.");
            } else {
                System.out.println(i + " was not found in the set.");
            }
        }
    }
}
```

```
1 was not found in the set.
2 was not found in the set.
3 was not found in the set.
4 was found in the set.
5 was not found in the set.
6 was not found in the set.
7 was found in the set.
8 was found in the set.
9 was not found in the set.
10 was not found in the set.
```

SECTION 21

Exercise 148

Getting an iterator.

```
import java.util.ArrayList;
import java.util.Iterator;
public class Main {
    public static void main(String[] args) {
        // Make a collection
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        // Get the iterator
        Iterator<String> it = cars.iterator();

        // Print the first item
        System.out.println(it.next());
    }
}
```

Volvo

Exercise 149

Looping through a collection.

```
import java.util.ArrayList;
import java.util.Iterator;
public class Main {
    public static void main(String[] args) {
        // Make a collection
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        // Get the iterator
        Iterator<String> it = cars.iterator();

        // Loop through a collection
        while(it.hasNext()) {
            System.out.println(it.next());
        }
    }
}
```

Volvo
BMW
Ford
Mazda

Exercise 150

Removing items from a collection.

```
import java.util.ArrayList;
import java.util.Iterator;
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<Integer>();
        numbers.add(12);
        numbers.add(8);
        numbers.add(2);
        numbers.add(23);
        Iterator<Integer> it = numbers.iterator();
        while(it.hasNext()) {
            Integer i = it.next();
            if(i < 10) {
                it.remove();
            }
        }
        System.out.println(numbers);
    }
}
```

[12, 23]

Exercise 151

Create an ArrayList that should store numbers (integers)

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> myNumbers = new ArrayList<Integer>();
        myNumbers.add(10);
        myNumbers.add(15);
        myNumbers.add(20);
        myNumbers.add(25);
        for (int i : myNumbers) {
            System.out.println(i);
        }
    }
}
```

**10
15
20
25**

Exercise 152

Create wrapper objects.

```
public class Main {
    public static void main(String[] args) {
        Integer myInt = 5;
        Double myDouble = 5.99;
        Character myChar = 'A';
        System.out.println(myInt);
        System.out.println(myDouble);
        System.out.println(myChar);
    }
}
```

**5
5.99
A**

Exercise 153

Using wrapper methods.

```
public class Main {
    public static void main(String[] args) {
        Integer myInt = 5;
        Double myDouble = 5.99;
        Character myChar = 'A';
        System.out.println(myInt.intValue());
        System.out.println(myDouble.doubleValue());
        System.out.println(myChar.charValue());
    }
}
```

```
5
5.99
A
```

Exercise 154

Convert wrapper objects to strings.

```
public class Main {
    public static void main(String[] args) {
        Integer myInt = 100;
        String myString = myInt.toString();
        System.out.println(myString.length());
    }
}
```

```
3
```

SECTION 22

Exercise 155

Create a program to use the Try... Catch statement.

```
public class Main {
    public static void main(String[] args) {
        try {
            int[] myNumbers = {1, 2, 3};
            System.out.println(myNumbers[10]);
        } catch (Exception e) {
            System.out.println("Something went wrong.");
        }
    }
}
```

```
Something went wrong.
```

Exercise 156

Create a program to use the Finally statement.

```
public class Main {
    public static void main(String[] args) {
        try {
            int[] myNumbers = {1, 2, 3};
            System.out.println(myNumbers[10]);
        } catch (Exception e) {
            System.out.println("Something went wrong.");
        } finally {
            System.out.println("The 'try catch' is finished.");
        }
    }
}
```

```
Something went wrong.
The 'try catch' is finished.
```

Exercise 157

Create a program to use the Throw statement.

```
public class Main {
    static void checkAge(int age) {
        if (age < 18) {
            throw new ArithmeticException("Access denied - You must be at least 18 years old.");
        } else {
            System.out.println("Access granted - You are old enough!");
        }
    }
    public static void main(String[] args) {
        checkAge(15);
    }
}
```

```
Exception in thread "main" java.lang.ArithmeticException: Access denied - You must be at least 18 years old.
    at Main.checkAge(Main.java:4)
    at Main.main(Main.java:12)
```

Exercise 158

Create a program to search for the word "Java" in a sentence.

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class Main {
    public static void main(String[] args) {
        Pattern pattern = Pattern.compile("Java", Pattern.CASE_INSENSITIVE);
        Matcher matcher = pattern.matcher("Java programming!");
        boolean matchFound = matcher.find();
        if (matchFound) {
            System.out.println("Match found");
        } else {
            System.out.println("Match not found");
        }
    }
}
```

Match found

Exercise 159

Create a program to run a thread by extending the thread class.

```
public class Main extends Thread {
    public static void main(String[] args) {
        Main thread = new Main();
        thread.start();
        System.out.println("This code is outside of the thread");
    }
    public void run() {
        System.out.println("This code is running in a thread");
    }
}
```

```
This code is outside of the thread
This code is running in a thread
```

Exercise 160

Create program to run a thread by implementing the Runnable interface.

```
public class Main implements Runnable {
    public static void main(String[] args) {
        Main obj = new Main();
        Thread thread = new Thread(obj);
        thread.start();
        System.out.println("This code is outside of the thread");
    }
    public void run() {
        System.out.println("This code is running in a thread");
    }
}
```

This code is outside of the thread
This code is running in a thread

Exercise 161

Create a program with concurrency problems.

```
public class Main extends Thread {
    public static int amount = 0;
    public static void main(String[] args) {
        Main thread = new Main();
        thread.start();
        System.out.println(amount);
        amount++;
        System.out.println(amount);
    }
    public void run() {
        amount++;
    }
}
```

0
2

Exercise 162

Create a program to avoid concurrency problems.

```
public class Main extends Thread {
    public static int amount = 0;
    public static void main(String[] args) {
        Main thread = new Main();
        thread.start();
        // Wait for the thread to finish
        while(thread.isAlive()) {
            System.out.println("Waiting...");
        }
        // Update amount and print its value
        System.out.println("Main: " + amount);
        amount++;
        System.out.println("Main: " + amount);
    }
    public void run() {
        amount++;
    }
}
```

Waiting...
Main: 1
Main: 2

SECTION 23

Exercise 163

Create a program to create a file.

```
import java.io.File;
import java.io.IOException;

public class CreateFile {
    public static void main(String[] args) {
        try {
            File myObj = new File("filename.txt");
            if (myObj.createNewFile()) {
                System.out.println("File created: " + myObj.getName());
            } else {
                System.out.println("File already exists.");
            }
        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

```
File created: filename.txt
```

Exercise 164

Create a program to write a file.

```
import java.io.FileWriter;
import java.io.IOException;

public class WriteToFile {
    public static void main(String[] args) {
        try {
            FileWriter myWriter = new FileWriter("filename.txt");
            myWriter.write("Files in Java might be tricky, but it is fun enough!");
            myWriter.close();
            System.out.println("Successfully wrote to the file.");
        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

```
Successfully wrote to the file.
```

Exercise 165

Create a program to read a file.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class ReadFile {
    public static void main(String[] args) {
        try {
            File myObj = new File("filename.txt");
            Scanner myReader = new Scanner(myObj);
            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                System.out.println(data);
            }
            myReader.close();
        } catch (FileNotFoundException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

Files in Java might be tricky, but it is fun enough!

Exercise 166

Create a program to get file information.

```
import java.io.File;

public class GetFileInfo {
    public static void main(String[] args) {
        File myObj = new File("filename.txt");
        if (myObj.exists()) {
            System.out.println("File name: " + myObj.getName());
            System.out.println("Absolute path: " + myObj.getAbsolutePath());
            System.out.println("Writeable: " + myObj.canWrite());
            System.out.println("Readable: " + myObj.canRead());
            System.out.println("File size in bytes: " + myObj.length());
        } else {
            System.out.println("The file does not exist.");
        }
    }
}
```

```
File name: filename.txt
Absolute path: C:\Users\MyName\filename.txt
Writeable: true
Readable: true
File size in bytes: 0
```

Exercise 167

Create a program to delete a file.

```
import java.io.File;

public class DeleteFile {
    public static void main(String[] args) {
        File myObj = new File("filename.txt");
        if (myObj.delete()) {
            System.out.println("Deleted the file: " + myObj.getName());
        } else {
            System.out.println("Failed to delete the file.");
        }
    }
}
```

Deleted the file: filename.txt