SWEN20003
Object Oriented Software Development

A Quick Tour of Java - Questions

Bach Le
bach.le@unimelb.edu.au

University of Melbourne

Learning Outcomes

Upon completion of this topic you will be able to:

- Identify some of the key Java features
- Understand the following in context of Java:
  - Identifiers, Data Types, Variables and Constants
  - Operators and Expressions
  - Flow of control
- Write simple Java programs

Which one of the following is **NOT** true?

1. Java is an Object Oriented language.
2. Java has a preprocessor to parse #define, #include type statements.
3. Java has no explicit pointer type.
4. Java code is portable across platforms.
5. JavaScript has no direct relationship to Java.

Which one of the following is **NOT** true?

1. Java is an Object Oriented language.
2. Java has a preprocessor to parse #define, #include type statements.
3. Java has no explicit pointer type.
4. Java code is portable across platforms.
5. JavaScript has no direct relationship to Java.

Answer:

2. Java has a preprocessor to parse #define, #include type statements.

A Java Applet has a `main` method. True or False?

Answer:

- True
- False

A Java Applet has a `main` method. True or False?

Answer:

- True
- False

Answer: False

In Java, the closest analogy to a C *function* is called a/an:

1. attribute
2. class
3. interface
4. method
5. object

In Java, the closest analogy to a C *function* is called a/an:

1. attribute
2. class
3. interface
4. method
5. object

Answer: 4. method

Select the correct answer by reviewing the following program.

```
1       HelloWorld.java: Hello World Application
2       #include java.lang.*;
3       public class HelloWorld {
4           public void main(int argv, String args*) {
5               System.out.println("Hello World!");
6           }
7       }
```

1. Only line 1 is incorrect.
2. Only lines 1 and 5 are incorrect.
3. Only lines 1, 2 and 4 are incorrect.
4. Only lines 2 and 5 are incorrect.
5. Only line 4 is incorrect.

Select the correct answer by reviewing the following program.

```
1       HelloWorld.java: Hello World Application
2       #include java.lang.*;
3       public class HelloWorld {
4           public void main(int argv, String args*) {
5               System.out.println("Hello World!");
6           }
7       }
```

1. Only line 1 is incorrect.
2. Only lines 1 and 5 are incorrect.
3. Only lines 1, 2 and 4 are incorrect.
4. Only lines 2 and 5 are incorrect.
5. Only line 4 is incorrect.

Answer:

3. Only lines 1, 2 and 4 are incorrect.

Can you fix the errors?

```
1           HelloWorld.java: Hello World Application
2           #include java.lang.*;
3           public class HelloWorld {
4               public void main(int argv, String args*) {
5                   System.out.println("Hello World!");
6               }
7           }
```

Can you fix the errors?

```
1           HelloWorld.java: Hello World Application
2           #include java.lang.*;
3           public class HelloWorld {
4               public void main(int argv, String args*) {
5                   System.out.println("Hello World!");
6               }
7           }
```

Line 1: Comment symbol missing

- `//HelloWorld.java: Hello World Application`

Can you fix the errors?

```
1           HelloWorld.java: Hello World Application
2           #include java.lang.*;
3           public class HelloWorld {
4               public void main(int argv, String args*) {
5                   System.out.println("Hello World!");
6               }
7           }
```

Line 1: Comment symbol missing

- `//HelloWorld.java: Hello World Application`

Line 2: #include instead of import

- `import java.lang.*;`

Can you fix the errors?

```
1           HelloWorld.java: Hello World Application
2           #include java.lang.*;
3           public class HelloWorld {
4               public void main(int argv, String args*) {
5                   System.out.println("Hello World!");
6               }
7           }
```

Line 1: Comment symbol missing

- `//HelloWorld.java: Hello World Application`

Line 2: #include instead of import

- `import java.lang.*;`

Line 4: Wrong signature for the main method

- `public static void main(String args[])` OR
- `public static void main(String[] args)`

```
1       public class CommandLineTest {
2           public static void main(String args[]) {
3               int count, i = 0;
4               count = args.length;
5               while(i < count) {
6                   System.out.println("Java is " + args[i] + "!");
7                   i = i + 1;
8               }
9           }
10      }
```

If you run:

`java CommandLineTest Simple Object-oriented Powerful`

the output will be:

```
1       public class CommandLineTest {
2           public static void main(String args[]) {
3               int count, i = 0;
4               count = args.length;
5               while(i < count) {
6                   System.out.println("Java is " + args[i] + "!");
7                   i = i + 1;
8               }
9           }
10      }
```

If you run:

`java CommandLineTest Simple Object-oriented Powerful`

the output will be:

Answer:

```
Java is Simple!
Java is Object-oriented!
Java is Powerful!
```

Which one of the following regarding Java *identifiers* is **incorrect**?

1. Must not start with a digit
2. The characters must be letters, digits, or underscore symbols
3. Can be of any length
4. Are not case-sensitive
5. Cannot be Java keywords

Which one of the following regarding Java *identifiers* is **incorrect**?

1. Must not start with a digit
2. The characters must be letters, digits, or underscore symbols
3. Can be of any length
4. Are not case-sensitive
5. Cannot be Java keywords

Answer:

(4) Are not case-sensitive

Which one of the following variable names is the best to choose for a variable to store the window height in a windowing application?

1. `window_height`
2. `_windowHeight`
3. `windowHeight`
4. `w`
5. `WindowHeight`

Which one of the following variable names is the best to choose for a variable to store the window height in a windowing application?

1. `window_height`
2. `_windowHeight`
3. `windowHeight`
4. `w`
5. `WindowHeight`

Answer:

`windowHeight`

Which one of the following is suitable to be used as a Java *identifier*?

1. `System`
2. `length`
3. `public`
4. `class`
5. `String`

Which one of the following is suitable to be used as a Java *identifier*?

1. `System`
2. `length`
3. `public`
4. `class`
5. `String`

Answer:

(2) `length`

Which one of the following statements is incorrect?

1. `int is a primitive data type.`
2. `String is a derived data type.`
3. `Array is a derived data type.`
4. `Float is a primitive data type.`
5. `boolean is a primitive data type.`

Which one of the following statements is incorrect?

1. `int is a primitive data type.`
2. `String is a derived data type.`
3. `Array is a derived data type.`
4. `Float is a primitive data type.`
5. `boolean is a primitive data type.`

Answer:

(4) `Float` is a primitive data type.

Which one of the assignment statements is **INCORRECT**?

1. `double a = 2;`
2. `double b = 2.7;`
3. `float c = 1.7;`
4. `double d = 2.70;`
5. `double e = 2/3;`

Which one of the assignment statements is **INCORRECT**?

1. `double a = 2;`
2. `double b = 2.7;`
3. `float c = 1.7;`
4. `double d = 2.70;`
5. `double e = 2/3;`

Answer:

(3) `float c = 1.7;`.

Which one of the assignment statements is **INCORRECT**?

1. `double a = 2;`
2. `double b = 2.7;`
3. `float c = 1.7;`
4. `double d = 2.70;`
5. `double e = 2/3;`

Answer:

(3) `float c = 1.7;`.

Correct assignment: `float c = 1.7F;`

Consider the following variable definitions.

```
float x = 1.2F; int y = 2; double z = 4.0;
char p = 'a'; short q = 1;
```

Which one of the assignment statements is **NOT** valid?

1. `y = q;`
2. `x = z;`
3. `x = y;`
4. `z = y;`
5. `x = p;`

Consider the following variable definitions.

```
float x = 1.2F; int y = 2; double z = 4.0;
char p = 'a'; short q = 1;
```

Which one of the assignment statements is **NOT** valid?

1. `y = q;`
2. `x = z;`
3. `x = y;`
4. `z = y;`
5. `x = p;`

Answer:
(2) `x = z;`

```
byte -> short -> int -> long -> float -> double
char -> int
```

Which of the following are variable types in Java? You may select more than one answer.

1. instance variable
2. local variable
3. method variable
4. global variable
5. static variable

Which of the following are variable types in Java? You may select more than one answer.

1. instance variable
2. local variable
3. method variable
4. global variable
5. static variable

Answer:

1, 2, 5

What is a *local variable* in Java?

What is a *local variable* in Java?

Answer:
A variable defined inside a Java method.

Which one of the following declarations is NOT valid?

1. `final int MAX_LENGTH = 420;`
2. `final double PI = 3.1428;`
3. `final char CHAR_CONST = "Z";`
4. `final boolean BOOL_CONST = true;`
5. `final String STR_CONST = "Hello World";`

Which one of the following declarations is NOT valid?

1. `final int MAX_LENGTH = 420;`
2. `final double PI = 3.1428;`
3. `final char CHAR_CONST = "Z";`
4. `final boolean BOOL_CONST = true;`
5. `final String STR_CONST = "Hello World";`

Answer:

(3) `final char CHAR_CONST = "Z";`

Consider the following Java program.

```
1  class CompareNumbers {
2      public static void main(String args[]) {
3          int a = 20,  b = 10;
4          boolean c = 0;
5          if (a < b)  c = 1;
6          System.out.println("c = " + c);
7      }
8  }
```

Which one of the following is correct?

1. The output will be: `c = 0`
2. The output will be: `c = 1`
3. The program will have a compilation error.
4. The program will have a run-time error.

Consider the following Java program.

```
1  class CompareNumbers {
2      public static void main(String args[]) {
3          int a = 20,  b = 10;
4          boolean c = 0;
5          if (a < b)  c = 1;
6          System.out.println("c = " + c);
7      }
8  }
```

Which one of the following is correct?

1. The output will be: `c = 0`
2. The output will be: `c = 1`
3. The program will have a compilation error.
4. The program will have a run-time error.

Answer:

(3) The program will have a compilation error.

Consider the following Java program.

```
1  class TwoWayExample {
2      public static void main(String args[]) {
3          int a = 20;
4          boolean b;
5          b = (a < 10) ? true : false;
6          System.out.println("b = " + b);
7      }
8  }
```

Which one of the following is correct?

1. The output will be: `b = true`
2. The output will be: `b = false`
3. The program will have a compilation error.
4. The program will have a run-time error.

Consider the following Java program.

```
1  class TwoWayExample {
2      public static void main(String args[]) {
3          int a = 20;
4          boolean b;
5          b = (a < 10) ? true : false;
6          System.out.println("b = " + b);
7      }
8  }
```

Which one of the following is correct?

1. The output will be: `b = true`
2. The output will be: `b = false`
3. The program will have a compilation error.
4. The program will have a run-time error.

Answer:

(2) The output will be: `b = false`

Write a C *function* that computes the average of two floating point numbers, your C program (`main`) can call.

Write a C *function* that computes the average of two floating point numbers, your C program (`main`) can call.

Answer:

```c
float calculate_average (float a, float b) {
    float result;
    result = (a + b)/2;
    return result;
}
```

Write a C *function* that computes the average of two floating point numbers, your C program (`main`) can call.

Answer:

```c
float calculate_average (float a, float b) {
    float result;
    result = (a + b)/2;
    return result;
}
```

Now write a Java *method* that computes the average of two floating point numbers, that your Java program (`main` method) can call.

Write a C *function* that computes the average of two floating point numbers, your C program (`main`) can call.

Answer:

```c
float calculate_average (float a, float b) {
    float result;
    result = (a + b)/2;
    return result;
}
```

Now write a Java *method* that computes the average of two floating point numbers, that your Java program (`main` method) can call.

Answer:

```c
static float calculateAverage (float a, float b) {
    float result;
    result = (a + b)/2;
    return result;
}
```

Write a Java method with the signature:

```
static int sumOfOdds(int start, int end)
```

which, given an odd number `start`, will return the sum of odd numbers from `start` to `end` (including `end` if it is an odd number).
e.g. `sumOfOdds(3,4)` will return 3, `sumOfOdds(3,7)` will return 15.

Write a Java method with the signature:

```
static int sumOfOdds(int start, int end)
```

which, given an odd number `start`, will return the sum of odd numbers from `start` to `end` (including `end` if it is an odd number).
e.g. `sumOfOdds(3,4)` will return 3, `sumOfOdds(3,7)` will return 15.

Answer:

```
1       static int sumOfOdds(int start, int end) {
2           int sum = start;
3           int next = start + 2;
4           while (next <= end) {
5               sum = sum + next;
6               next += 2;
7           }
8           return sum;
9       }
```

Write a Java method with the signature:

```
static int sumOfOddsN(int start, int n)
```

which, given an odd number `start`, will return the sum of the `n` ($n \geq 1$) odd numbers (including `start`) from `start` to `end`.
e.g. `sumOfOddsN(3,2)` will return 4 (3+5), `sumOfOddsN(3,3)` will return 15 (3+5+7).

Write a Java method with the signature:

```
static int sumOfOddsN(int start, int n)
```

which, given an odd number `start`, will return the sum of the `n` ($n \geq 1$) odd numbers (including `start`) from `start` to end.
e.g. `sumOfOddsN(3,2)` will return 4 (3+5), `sumOfOddsN(3,3)` will return 15 (3+5+7).

Answer:

```
1       static int sumOfOddsN(int start, int n) {
2           int sum = 0;
3           for (int i = 0; i < n; i++) {
4               sum += start;
5               start += 2;
6           }
7           return sum;
8       }
```

What will be the output of the following program?

```
public class BreakExample {
    public static void main(String[] args) {
        loop1: for (int i = 0; i < 3; i++) {
            loop2: for (int j = 0; j < 3; j++) {
                System.out.println("i=" + i + " j=" +j);
                if (j == 1)
                    break loop1;
            }
        }
    }
}
```

What will be the output of the following program?

```
public class BreakExample {
    public static void main(String[] args) {
        loop1: for (int i = 0; i < 3; i++) {
            loop2: for (int j = 0; j < 3; j++) {
                System.out.println("i=" + i + " j=" +j);
                if (j == 1)
                    break loop1;
            }
        }
    }
}
```

Answer:

```
        i=0 j=0
        i=0 j=1
```

What will be the output of the following program?

```
public class ContinueExample {
    public static void main(String[] args) {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (j == 1)
                    continue;
                System.out.println("i=" + i + " j=" +j);
            }
        }
    }
}
```

Answer:

```
i=0 j=0
i=0 j=2
i=1 j=0
i=1 j=2
i=2 j=0
i=2 j=2
```

Learning Outcomes

Upon completion of this topic you will be able to:

- Identify some of the key Java features
- Understand the following in context of Java:
  - Identifiers, Data Types, Variables and Constants
  - Operators and Expressions
  - Flow of control
- Write simple Java programs