

Q. What will be the result of attempting to compile and run the following class?

```
public class IfTest {  
    public static void main(String[] args) {  
        if (true)  
        if (false)  
        System.out.println("a");  
        else  
        System.out.println("b");  
    }  
}
```

Select the one correct answer.

- (a) The code will fail to compile because the syntax of the if statement is incorrect.
- (b) The code will fail to compile because the compiler will not be able to determine which if statement the else clause belongs to.
- (c) The code will compile correctly and display the letter a , when run.
- (d) The code will compile correctly and display the letter b , when run.
- (e) The code will compile correctly, but will not display any output.

Q. Which statements are true?

Select the three correct answers.

- (a) The conditional expression in an if statement can have method calls.
- (b) If a and b are of type boolean , the expression (a = b) can be the conditional expression of an if statement.
- (c) An if statement can have either an if clause or an else clause.
- (d) The statement if (false) ; else ; is illegal.
- (e) Only expressions which evaluate to a boolean value can be used as the condition in an if statement.

Q. What, if anything, is wrong with the following code?

```
void test(int x) {  
    switch (x) {  
        case 1:  
        case 2:  
        case 0:  
        default:  
        case 4: } }
```

Select the one correct answer.

- a)The variable x does not have the right type for a switch expression.

- d)The case label 0 must precede case label 1 .
- c)Each case section must end with a break statement.
- d)The default label must be the last label in the switch statement.
- e)The body of the switch statement must contain at least one statement.
- f)There is nothing wrong with the code

Q. Which of these combinations of switch expression types and case label value types are legal within a switch statement?

Select the two correct answers.

1. switch expression of type int and case label value of type char .
2. switch expression of type float and case label value of type int .
3. switch expression of type byte and case label value of type float .
4. switch expression of type char and case label value of type long .
5. switch expression of type boolean and case label value of type boolean .
6. switch expression of type Byte and case label value of type byte .
7. switch expression of type byte and case label value of type Byte .

Q. What will be the result of attempting to compile and run the following program?

```
public class Switching {
    public static void main(String[] args) {
        final int iLoc = 3;
        switch (6) {
            case 1:
            case iLoc:
            case 2 * iLoc:
                System.out.println("I am not OK.");
            default:
                System.out.println("You are OK.");
            case 4:
                System.out.println("It's OK.");
        }
    }
}
```

Select the one correct answer.

- (a) The code will fail to compile because of the case label value $2 * iLoc$.
- (b) The code will fail to compile because the default label is not specified last in the switch statement.
- (c) The code will compile correctly and will only print the following, when run:
I am not OK.

You are OK.

It's OK.

(d) The code will compile correctly and will only print the following, when run:

You are OK.

It's OK.

(e) The code will compile correctly and will only print the following, when run:

It's OK.

Q. What will be the result of attempting to compile and run the following program?

```
public class MoreSwitching {  
    public static void main(String[] args) {  
        final int iLoc = 3;  
        Integer iRef = 5;  
        switch (iRef) {  
            default:  
                System.out.println("You are OK.");  
            case 1:  
            case iLoc:  
            case 2 * iLoc:  
                System.out.println("I am not OK.");  
                break;  
            case 4:  
                System.out.println("It's OK.");  
        }  
    }  
}
```

Select the one correct answer.

(a) The code will fail to compile because the type of the switch expression is not valid.

(b) The code will compile correctly and will only print the following, when run:

You are OK.

I am not OK.

(c) The code will compile correctly and will only print the following, when run:

You are OK.

I am not OK.

It's OK.

(d) The code will compile correctly and will only print the following, when run:

It's OK.

Q. What will be the result of attempting to compile and run the following program?

```
public class KeepOnSwitching {  
    public static void main(String[] args) {  
        final int iLoc = 3;  
        final Integer iFour = 4;  
        Integer iRef = 4;  
        switch (iRef) {  
            case 1:  
            case iLoc:  
            case 2 * iLoc:  
                System.out.println("I am not OK.");  
            default:  
                System.out.println("You are OK.");  
            case iFour:  
                System.out.println("It's OK.");  
        }  
    }  
}
```

Select the one correct answer.

- (a) The code will fail to compile because of the value of one of the case labels.
- (b) The code will fail to compile because of the type of the switch expression.
- (c) The code will compile correctly and will only print the following, when run:
You are OK.
It's OK.
- (d) The code will compile correctly and will only print the following, when run:
It's OK.

Q.

Which digits, and in what order, will be printed when the following program is run?

```
public class MyClass {  
    public static void main(String[] args) {  
        int k=0;  
        try {  
            int i = 5/k;  
        } catch (ArithmeticException e) {  
            System.out.println("1");  
        } catch (RuntimeException e) {  
            System.out.println("2");  
            return;  
        } catch (Exception e) {  
            System.out.println("3");  
        } finally {  
            System.out.println("4");  
        }  
        System.out.println("5");  
    }  
}
```

Select the one correct answer.

- (a) The program will only print 5.
- (b) The program will only print 1 and 4, in that order.
- (c) The program will only print 1, 2, and 4, in that order.
- (d) The program will only print 1, 4, and 5, in that order.
- (e) The program will only print 1, 2, 4, and 5, in that order.
- (f) The program will only print 3 and 5, in that order.

Q.

Given the following program, which statements are true?

```
public class Exceptions {  
    public static void main(String[] args) {  
        try {  
            if (args.length == 0) return;  
            System.out.println(args[0]);  
        } finally {  
            System.out.println("The end");  
        }  
    }  
}
```

Select the two correct answers.

- (a) If run with no arguments, the program will produce no output.
- (b) If run with no arguments, the program will print "The end".
- (c) The program will throw an `ArrayIndexOutOfBoundsException`.
- (d) If run with one argument, the program will simply print the given argument.
- (e) If run with one argument, the program will print the given argument followed by "The end".

Q.

What will be the result of attempting to compile and run the following program?

```
public class MyClass {  
    public static void main(String[] args) {  
        RuntimeException re = null;  
        throw re;  
    }  
}
```

Select the one correct answer.

- (a) The code will fail to compile because the `main()` method does not declare that it throws `RuntimeException` in its declaration.
- (b) The program will fail to compile because it cannot throw `re`.
- (c) The program will compile without error and will throw `java.lang.RuntimeException` when run.
- (d) The program will compile without error and will throw `java.lang.NullPointerException` when run.
- (e) The program will compile without error and will run and terminate without any output.

Q.

Which statements are true?

Select the two correct answers.

- (a) If an exception is not caught in a method, the method will terminate and normal execution will resume.
- (b) An overriding method must declare that it throws the same exception classes as the method it overrides.
- (c) The `main()` method of a program can declare that it throws checked exceptions.
- (d) A method declaring that it throws a certain exception class may throw instances of any subclass of that exception class.
- (e) `finally` blocks are executed if, and only if, an exception gets thrown while inside the corresponding `try` block.

Q.

Which digits, and in what order, will be printed when the following program is compiled and run?

```
public class MyClass {
    public static void main(String[] args) {
        try {
            f();
        } catch (InterruptedException e) {
            System.out.println("1");
            throw new RuntimeException();
        } catch (RuntimeException e) {
            System.out.println("2");
            return;
        } catch (Exception e) {
            System.out.println("3");
        } finally {
            System.out.println("4");
        }
        System.out.println("5");
    }

    // InterruptedException is a direct subclass of Exception.
    static void f() throws InterruptedException {
        throw new InterruptedException("Time for lunch.");
    }
}
```

Select the one correct answer.

- (a) The program will print 5.
- (b) The program will print 1 and 4, in that order.
- (c) The program will print 1, 2, and 4, in that order.
- (d) The program will print 1, 4, and 5, in that order.
- (e) The program will print 1, 2, 4, and 5, in that order.
- (f) The program will print 3 and 5, in that order.

Q.

Which digits, and in what order, will be printed when the following program is run?

```
public class MyClass {
    public static void main(String[] args) throws InterruptedException {
        try {
            f();
            System.out.println("1");
        } finally {
            System.out.println("2");
        }

        System.out.println("3");
    }

    // InterruptedException is a direct subclass of Exception.
    static void f() throws InterruptedException {
        throw new InterruptedException("Time to go home.");
    }
}
```

Select the one correct answer.

- (a) The program will print 2 and throw InterruptedException.
- (b) The program will print 1 and 2, in that order.
- (c) The program will print 1, 2, and 3, in that order.
- (d) The program will print 2 and 3, in that order.
- (e) The program will print 3 and 2, in that order.
- (f) The program will print 1 and 3, in that order.

Q.

What is wrong with the following code?

```
public class MyClass {
    public static void main(String[] args) throws A {
        try {
            f();
        } finally {
            System.out.println("Done.");
        } catch (A e) {
            throw e;
        }
    }

    public static void f() throws B {
        throw new B();
    }
}

class A extends Throwable {}

class B extends A {}
```

Select the one correct answer.

- (a) The main() method must declare that it throws B.
- (b) The finally block must follow the catch block in the main() method.
- (c) The catch block in the main() method must declare that it catches B rather than A.
- (d) A single try block cannot be followed by both a finally and a catch block.
- (e) The declaration of class A is illegal.

Q.

What is the minimal list of exception classes that the overriding method f() in the following code must declare in its throws clause before the code will compile correctly?

```
class A {
    // InterruptedException is a direct subclass of Exception.
    void f() throws ArithmeticException, InterruptedException {
```

```

        div(5, 5);
    }

    int div(int i, int j) throws ArithmeticException {
        return i/j;
    }
}

public class MyClass extends A {
    void f() /* throws [...list of exceptions...] */ {
        try {
            div(5, 0);
        } catch (ArithmeticException e) {
            return;
        }
        throw new RuntimeException("ArithmeticException was expected.");
    }
}

```

Select the one correct answer.

- (a) Does not need to specify any exceptions.
- (b) Needs to specify that it throws ArithmeticException.
- (c) Needs to specify that it throws InterruptedException.
- (d) Needs to specify that it throws RuntimeException.
- (e) Needs to specify that it throws both ArithmeticException and InterruptedException.

Q.

What, if anything, would cause the following code not to compile?

```
class A {
    void f() throws ArithmeticException {
        //...
    }
}

public class MyClass extends A {
    public static void main(String[] args) {
        A obj = new MyClass();

        try {
            obj.f();
        } catch (ArithmeticException e) {
            return;
        } catch (Exception e) {
            System.out.println(e);
            throw new RuntimeException("Something wrong here");
        }
    }

    // InterruptedException is a direct subclass of Exception.
    void f() throws InterruptedException {
        //...
    }
}
```

Select the one correct answer.

- (a) The main() method must declare that it throws RuntimeException.
- (b) The overriding f() method in MyClass must declare that it throws ArithmeticException, since the f() method in class A declares that it does.
- (c) The overriding f() method in MyClass is not allowed to throw InterruptedException, since the f() method in class A does not throw this exception.
- (d) The compiler will complain that the catch(ArithmeticException) block shadows the catch(Exception) block.
- (e) You cannot throw exceptions from a catch block.
- (f) Nothing is wrong with the code, it will compile without errors.

