**Question 1**

Which of the following are command-line switches used to enable assertions in non-system classes?

|  |  |
| --- | --- |
| a. | -ea |
| b. | -ae |
| c. | -aon |
| d. | -aoff |
| e. | -enableassertions |
| f. | -assertionsenable |
| g. | -assertionson |
| h. | -assertionsoff |

**Question 2**

Which of the following are command-line switches used to disable assertions in non-system classes?

|  |  |
| --- | --- |
| a. | -da |
| b. | -ad |
| c. | -aon |
| d. | -aoff |
| e. | -disableassertions |
| f. | -assertionsdisable |
| g. | -assertionson |
| h. | -assertionsoff |

**Question 3**

Which of the following are command-line switches used to disable assertions in non-system classes?

|  |  |
| --- | --- |
| a. | -disableassertions |
| b. | -disableAssertions |
| c. | -assertionsDisable |
| d. | -assertionsOn |
| e. | -assertionsOff |
| f. | None of the above |

**Question 4**

Which of the following are command-line switches used to enable assertions in system classes?

|  |  |
| --- | --- |
| a. | -saon |
| b. | -saoff |
| c. | -esa |
| d. | -sae |
| e. | -systemassertionson |
| f. | -systemassertionsoff |
| g. | -enablesystemassertions |
| h. | -systemassertionsenable |

**Question 5**

Which of the following statements are true?

|  |  |
| --- | --- |
| a. | By default assertions are disabled at run-time. |
| b. | Assertions can be selectively enabled for any named class. |
| c. | If assertions are selectively enabled for a named class then assertions are automatically enabled for any subclass of the named class. |
| d. | Assertions can be selectively enabled for any named package. |
| e. | If assertions are selectively enabled for any named package then assertions are automatically enabled for the subpackages. |
| f. | Assertions can be selectively enable for the unnamed package in the current working directory. |
| g. | Assertions can be selectively enable for any unnamed package in any named directory. |

**Question 6**

Which of the following is thrown to indicate that an assertion has failed?

|  |  |
| --- | --- |
| a. | AssertError |
| b. | AssertException |
| c. | AssertionError |
| d. | AssertionException |
| e. | None of the above |

**Question 7**

class A {

void m1(int i) {

int j = i % 3;

switch (j) {

case 0: System.out.print("0"); break;

case 1: System.out.print("1"); break;

default:

assert j == 2;

System.out.print(j);

}}

public static void main (String[] args) {

A a = new A();

for (int i=5; i >= -1; i--) {a.m1(i);}

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints 210210-1 followed by an AssertionError message. |
| b. | With assertions enabled it prints 210210 followed by an AssertionError message. |
| c. | With assertions enabled it prints only 210210. |
| d. | With assertions enabled it prints nothing. |
| e. | With assertions disabled it prints 210210-1 |
| f. | With assertions disabled it prints only 210210 |
| g. | Assertions should not be used within the default case of a switch statement. |

**Question 8**

class B {

private static int x1;

public void setX(int x) {x1 = x;}

public int getX() {return x1;}

public static void main(String[] args) {

int i1 = 0, j1 = 0;

do {

System.out.print(j1++);

assert (i1 = j1 + x1) < 6;

} while (i1 < 3);

}}

Assuming that the setX method is never invoked, which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints 012345 followed by an AssertionError message. |
| b. | With assertions enabled it prints 012. |
| c. | With assertions disabled it prints: 012345 |
| d. | With assertions disabled it prints: 012 |
| e. | With assertions disabled it attempts to print an infinite sequence of numbers. |
| f. | With assertions disabled the variable i1 is incremented with each pass through the loop. |
| g. | As a rule, the boolean expression of an assert statement should not be used to perform actions that are required for normal operation of the program. |

**Question 9**

Which statements are true?

|  |  |
| --- | --- |
| a. | Assertions should not be used to validate arguments passed to public methods. |
| b. | Assertions should not be used to validate arguments passed to nonpublic methods. |
| c. | Assertions should not be used within the default case of a switch statement. |
| d. | Assertions should not be used to perform processing that is required for the normal operation of the application. |

**Question 10**

Which statements are true?

|  |  |
| --- | --- |
| a. | Assertions should never be placed at any point in the code that should not be reached under normal circumstances. |
| b. | The compiler will generate an error if an assert statement is "unreachable" as defined by the Java Language Specification. |
| c. | A catch clause should not be used to catch an AssertionError. |
| d. | AssertionError is a checked exception. |

**Question 11**

class C {

String m1(int i) {

switch (i) {

case 0: return "A";

case 1: return "B";

case 2: return "C";

default: throw new AssertionError();

}}

public static void main(String[] args) {

C c = new C();

for (int i = 0; i < 4; i++) {

System.out.print(c.m1(i));

}}}

Which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints ABC followed by an AssertionError message. |
| b. | With assertions disabled it prints ABC followed by an AssertionError message. |
| c. | Assertions should not be used within the default case of a switch statement. |
| d. | In this code example an assert statement could not be used in place of the "throw" statement. |

**Question 12**

class C {

String m1(int i) {

switch (i) {

case 0: return "A";

case 1: return "B";

case 2: return "C";

default:

assert false;

}

return "E";

}

public static void main(String[] args) {

C c = new C();

for (int i = 0; i < 4; i++) {

System.out.print(c.m1(i));

}}}

Which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints ABC followed by an AssertionError message. |
| b. | With assertions enabled it prints ABCE followed by an AssertionError message. |
| c. | With assertions disabled it prints ABC |
| d. | With assertions disabled it prints ABCE |
| e. | Assertions should not be used within the default case of a switch statement. |

**Question 13**

class C {

String m1(int i) {

switch (i) {

case 0: return "A";

case 1: return "B";

case 2: return "C";

default:

assert false;

}

}

public static void main(String[] args) {

C c = new C();

for (int i = 0; i < 4; i++) {

System.out.print(c.m1(i));

}}}

Which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints ABC followed by an AssertionError message. |
| b. | With assertions disabled it prints ABC followed by an AssertionError message. |
| c. | Assertions should not be used within the default case of a switch statement. |
| d. | In this code example a throw statement must be used in place of the assert statement. |
| e. | Compile-time error |

**Question 14**

class D {

private boolean b1, b2;

public void setB1(boolean b) {b1 = b;}

public void setB2(boolean b) {b2 = b;}

public void m1 () {

if (!b2 & !b1) {System.out.print("A");

} else if (!b2 & b1) {System.out.print("B");

} else if (b2 & !b1) {System.out.print("C");

} else {assert false;}

}

public static void main (String[] args) {

D d = new D();

d.setB1(true); d.setB2(true);

d.m1();

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints an AssertionError message. |
| b. | With assertions enabled it prints nothing. |
| c. | With assertions disabled it prints an AssertionError message. |
| d. | With assertions disabled it prints nothing. |
| e. | An assertion should not be placed at any location that the programmer believes will never be reached under normal operating conditions. |
| f. | The assert statement is being used to check a control-flow invariant to verify that the control flow never reaches a point in the program. |

**Question 15**

class A {

private void m1 (int i) {

assert i < 10 : i; System.out.print(i);

}

public void m2 (int i) {

assert i < 10 : i; System.out.print(i);

}

public static void main (String[] args) {

A a = new A(); a.m1(11); a.m2(12);

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | If assertions are enabled at run time it prints an error message. |
| b. | With assertions enabled it prints nothing. |
| c. | With assertions disabled it prints an error message. |
| d. | With assertions disabled it prints 1112. |
| e. | With assertions disabled it prints nothing. |
| f. | The assert statements are being used to check a precondition--something that must be true when the method is invoked. |
| g. | Method m1 is an example of an improper use of an assert statement: an assert statement should not be used for argument checking in a non-public method. |
| h. | Method m2 is an example of an improper use of an assert statement: an assert statement should not be used for argument checking in a public method. |

**Question 16**

class B {

int a, b, c;

private void setA(int i) {a = i;}

private void setB(int i) {b = i;}

private int m1 (int i) {

c = a + b + i; assert c < 200 : c; return c;

}

public static void main (String[] args) {

B b = new B(); b.setA(50); b.setB(100); b.m1(50);

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | If assertions are not enabled at run time it prints an error message. |
| b. | If assertions are not enabled at run time it prints nothing. |
| c. | With assertions enabled it prints an error message. |
| d. | With assertions enabled it prints nothing. |
| e. | The assert statement is being used to check a postcondition--something that must be true when the method completes successfully. |

**Question 17**

class C {

int a, b, c;

public void setA(int i) {a = i; assert validateC() : c;}

public void setB(int i) {b = i; assert validateC() : c;}

private boolean validateC() {

return c > a + 2 \* b;

}

public int m1(int i) {

c = a + b + i;

assert validateC() : c;

return c;

}

public C(int i) {

c = i; assert validateC() : c;

}

public static void main(String[] args) {

C c = new C(251); c.setA(50); c.setB(100);

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | If assertions are not enabled at run time it prints an error message. |
| b. | If assertions are not enabled at run time it prints nothing. |
| c. | With assertions enabled it prints an error message. |
| d. | With assertions enabled it prints nothing. |
| e. | The assert statement is being used to check a class invariant--something that must be true about each instance of the class. |
| f. | The assert statements are being used to check a precondition--something that must be true when the method is invoked. |

**Question 18**

class E {

private boolean b1, b2, b3;

public void setB1(boolean b) {b1 = b;}

public void setB2(boolean b) {b2 = b;}

public void setB3(boolean b) {b3 = b;}

public void m1 (int i) {

b2 = i % 2 == 0;

if (!b3 & !b2 & !b1) {System.out.print("A");

} else if (!b3 & !b2 & b1) {System.out.print("B");

} else if (!b3 & b2 & !b1) {System.out.print("C");

} else { // Only b3 is true.

assert b3 & !b2 & !b1;

}

System.out.print(b1 + "," + b2 + "," + b3);

b1 = b2 = b3 = false;

}

public static void main (String[] args) {

E e = new E(); e.setB1(true); e.m1(2);

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | With assertions enabled it prints an error message. |
| b. | With assertions enabled it prints: true,true,false |
| c. | With assertions disabled it prints an error message. |
| d. | With assertions disabled it prints: true,true,false |
| e. | With assertions disabled it prints nothing. |
| f. | The combination of the if/else statements and the assert statement indicate that the programmer expects no more than one boolean, b1, b2 or b3, to be true. |
| g. | The assert statement is being used to check a precondition--something that must be true when the method begins. |
| h. | The assert statement is being used to check an internal invariant--something that the programmer assumes to be true at a particular point in the program. |

**Question 19**

class F {

private boolean b1, b2, b3;

public void setB1(boolean b) {b1 = b;}

public void setB2(boolean b) {b2 = b;}

public void setB3(boolean b) {b3 = b;}

public String m1 (int i) {

b2 = i % 2 == 0;

if (!b3 & !b2 & !b1) {return "None are true.";

} else if (!b3 & !b2 & b1) {return "Only b1 is true.";

} else if (!b3 & b2 & !b1) {return "Only b2 is true.";

} else if (b3 & !b2 & !b1) {return "Only b3 is true.";

} else {throw new AssertionError();}

}

public static void main (String[] args) {

F f = new F();

f.setB1(true);

System.out.println(f.m1(5));

System.out.println(f.m1(6));

}}

Which statements are true?

|  |  |
| --- | --- |
| a. | Prints "Only b1 is true" followed by an error message. |
| b. | An assertion should not be placed at any location that the programmer believes will never be reached under normal operating conditions. |
| c. | The combination of the if/else statements and the assert statement indicate that the programmer expects no more than one boolean, b1, b2, or b3, to be true. |
| d. | The assert statement is being used to check a precondition--something that must be true when the method begins. |
| e. | The assert statement is being used to check a control-flow invariant to verify that the control flow never reaches a point in the program. |
| f. | The throw statement could be replaced by an assert statement. |

**Question 20**

An assert statement can be used to check a control-flow invariant to verify which of the following?

|  |  |
| --- | --- |
| a. | A particular assumption is true when the flow of control enters a method. |
| b. | The flow of control does not reach a particular point in the program. |
| c. | The normal flow of control has reached a particular point in the program. |
| d. | The normal flow of control has reached the end of a method. |
| e. | The default case of a switch statement is not reached. |
| f. | The else block of an if/else statement is not reached. |