1. **Consider the following code snippet:**

```
if(i == 10.0)
System.out.println("true");
```

Which one of the following declarations of the variable i will compile without errors and print true when the program executes?

a) int i = 012;
b) int i = 10.0f;
c) int i = 10L;
d) int i = 10.0;

2. **Consider the following program:**

```
import java.math.BigDecimal;
class NumberTest {
public static void main(String []args) {
Number [] numbers = new Number[4];
numbers[0] = new Number(0); // NUM
numbers[1] = new Integer(1);
numbers[2] = new Float(2.0f);
numbers[3] = new BigDecimal(3.0); // BIG
for(Number num : numbers) {
System.out.print(num + " ");
}
}
}
```

Which one of the following options correctly describes the behavior of this program?

a) Compiler error in line marked with comment NUM because Number cannot be instantiated.
b) Compiler error in line marked with comment BIG because BigDecimal does not inherit from Number.
c) When executed, this program prints the following: 0 1 2.0 3.
d) When executed, this program prints the following: 0.0 1.0 2.0 3.0.

3. **Consider the following code segment:**

```
StringBuffer strBuffer = new StringBuffer("This, that, etc.!");
System.out.println(strBuffer.replace(12, 15, "etcetera"));
```

Which one of the following options correctly describes the behavior of this code segment?

a) This code segment: This, that, etcetera.!
b) This code segment: This, that, etcetera!
c) This code segment: This, that, etc.
d) This program throws in an ArrayIndexOutOfBoundsException.

4. **Consider the following program:**

```
class SBAppend {
public static void main(String []args) {
Object nullObj = null;
StringBuffer strBuffer = new StringBuffer(10);
strBuffer.append("hello ");
strBuffer.append("world ");
strBuffer.append(nullObj);
strBuffer.insert(11, '!');
System.out.println(strBuffer);
}
}
```

Which one of the following options correctly describes the behavior of this program?

a) This program prints the following: hello world!

b) This program prints the following: hello world! null

c) This program throws a NullPointerException.

d) This program throws an InvalidArgumentException.

e) This program throws an ArrayIndexOutOfBoundsException.

5. **Consider the following code segment:**

```
Boolean b = null;
System.out.println(b ? true : false);
```

Which one of the following options correctly describes the behavior of this code segment?

a) This code will result in a compiler error since a reference type (of type Boolean) cannot be used as part of expression for condition check.

b) This code will result in a throwing a NullPointerException.

c) This code will print true in console.

d) This code will print false in console.

6. **What will be the output of the following program?**

```
class Base {
public Base() {
System.out.println("Base");
}
}
class Derived extends Base {
public Derived() {
System.out.println("Derived");
}
}
class DeriDerived extends Derived {
public DeriDerived() {
System.out.println("DeriDerived");
}
}
class Test {
public static void main(String []args) {
Derived b = new DeriDerived();
}
}
```

a) Base
Derived
DeriDerived

b) Derived
DeriDerived

c) DeriDerived
Derived
Base

d) DeriDerived
Derived

7. **Consider the following code segment:**

```
MODIFIER class SomeClass { }
```

Which three of the following modifiers, when replaced instead of MODIFIER, will compile cleanly?

a) public
b) protected
c) private
d) abstract
e) final
f) static

8. **Consider the following class definition:**

```
class Point {
private int x = 0, y;
public Point(int x, int y) {
this.x = x;
this.y = y;
}
// DEFAULT_CTOR
}
```

Which one of the following definitions of the Point constructor can be replaced without compiler errors in place of the comment DEFAULT_CTOR?

```
a) public Point() {
this(0, 0);
super();
}
b) public Point() {
super();
this(0, 0);
}
c) private Point() {
this(0, 0);
}
d) public Point() {
this();
}
e) public Point() {
this(x, 0);
}
```

9. **Consider the following program:**

```
class Base {
public Base() {
System.out.print("Base ");
}
public Base(String s) {
System.out.print("Base: " + s);
}
}
class Derived extends Base {
public Derived(String s) {
super(); // Stmt-1
super(s); // Stmt-2
System.out.print("Derived ");
}
}
class Test {
public static void main(String []args) {
Base a = new Derived("Hello ");
}
}
```

Select three correct options from the following list:

a) Removing Stmt-1 will make the program compilable and it will print the following: Base Derived.

b) Removing Stmt-1 will make the program compilable and it will print the following: Base: Hello Derived.

c) Removing Stmt-2 will make the program compilable and it will print the following: Base Derived.

d) Removing both Stmt-1 and Stmt-2 will make the program compilable and it will print the following: Base Derived.

e) Removing both Stmt-1 and Stmt-2 will make the program compilable and it will print the following: Base: Hello Derived.

10. **You want to use the static member MYCONST belonging to class A in abc.org.project package. Which one of the following statements shows the correct use of static import feature?**

a) static import abc.org.project.A;

b) static import abc.org.project.A.MYCONST;

c) import static abc.org.project.A;

d) import static abc.org.project.A.MYCONST;

11. **Which one of the following programs compiles without any errors and prints “hello world” in console?**

```
a) import static java.lang.System.out.println;
class StaticImport {
public static void main(String []args) {
println("hello world");
}
}
```

```
b) import static java.lang.System.out;
class StaticImport {
public static void main(String []args) {
out.println("hello world");
}
}
```

```
c) import static java.lang.System.out.*;
class StaticImport {
public static void main(String []args) {
out.println("hello world");
}
}
```

```
d) import static java.lang.System.out.*;
class StaticImport {
public static void main(String []args) {
println("hello world");
}
}
```

12. **Consider the following program and choose the right option from the given list:**

```
class Base {
public void test() {
protected int a = 10; // #1
}
}
class Test extends Base { // #2
public static void main(String[] args) {
System.out.printf(null); // #3
}
}
```

a) The compiler will report an error at statement #1.
b) The compiler will report an error at statement #2.
c) The compiler will report errors at statement #3.
d) The program will compile without any error.

13. **Consider the following program and choose the correct option from the list of options:**

```
class Base {
public void test() {}
}
class Base1 extends Base {
public void test() {
System.out.println("Base1");
}
}
class Base2 extends Base {
public void test() {
System.out.println("Base2");
}
}
class Test {
public static void main(String[] args) {
Base obj = new Base1();
((Base2)obj).test(); // CAST
}
}
```

a) The program will print the following: Base1.
b) The program will print the following: Base2.
c) The compiler will report an error in the line marked with comment CAST.
d) The program will result in an exception (ClassCastException).

14. **Consider the following program:**

```
class Outer {
class Inner {
public void print() {
System.out.println("Inner: print");
}
}
}
class Test {
public static void main(String []args) {
// Stmt#1
inner.print();
}
}
```

Which one of the following statements will you replace in place of // Stmt#1 to make the program compile and run successfully to print “Inner: print” in console?

a) Outer.Inner inner = new Outer.Inner();
b) Inner inner = new Outer.Inner();
c) Outer.Inner inner = new Outer().Inner();
d) Outer.Inner inner = new Outer().new Inner();

15. **Consider the following program:**

```
public class Outer {
private int mem = 10;
class Inner {
private int imem = new Outer().mem; // ACCESS1
}
public static void main(String []s) {
System.out.println(new Outer().new Inner().imem); // ACCESS2
}
}
```

Which one of the following options is correct?

a) When compiled, this program will result in a compiler error in line marked with comment ACCESS1.
b) When compiled, this program will result in a compiler error in line marked with comment ACCESS2.
c) When executed, this program prints 10.
d) When executed, this program prints 0.

16. **Consider the following program:**

```
interface EnumBase { }
enum AnEnum implements EnumBase { // IMPLEMENTS_INTERFACE
ONLY_MEM;
}
class EnumCheck {
public static void main(String []args) {
if(AnEnum.ONLY_MEM instanceof AnEnum) {
System.out.println("yes, instance of AnEnum");
}
if(AnEnum.ONLY_MEM instanceof EnumBase) {
System.out.println("yes, instance of EnumBase");
}
if(AnEnum.ONLY_MEM instanceof Enum) { // THIRD_CHECK
System.out.println("yes, instance of Enum");
}
}
}
```

Which one of the following options is correct?

a) This program results in a compiler in the line marked with comment IMPLEMENTS_INTERFACE.
b) This program results in a compiler in the line marked with comment THIRD_CHECK.
c) When executed, this program prints the following:

```
yes, instance of AnEnum
```

d) When executed, this program prints the following:

```
yes, instance of AnEnum
yes, instance of EnumBase
```

e) When executed, this program prints the following:

```
yes, instance of AnEnum
yes, instance of EnumBase
yes, instance of Enum
```

17. **Which of the following statements are true with respect to enums? (Select all that apply.)**

a) An enum can have private constructor.
b) An enum can have public constructor.
c) An enum can have public methods and fields.
d) An enum can implement an interface.
e) An enum can extend a class.

18. **Consider the following program and predict the behavior:**

```
class base1 {
protected int var;
}
interface base2 {
int var = 0; // #1
}
class Test extends base1 implements base2 { // #2
public static void main(String args[]) {
System.out.println("var:" + var); // #3
}
}
```

a) The program will report a compilation error at statement #1.
b) The program will report a compilation error at statement #2.
c) The program will report a compilation error at statement #3.
d) The program will compile without any errors.

19. **Consider the following program:**

```
class WildCard {
interface BI {}
interface DI extends BI {}
interface DDI extends DI {}
static class C<T> {}
static void foo(C<? super DI> arg) {}
public static void main(String []args) {
foo(new C<BI>()); // ONE
foo(new C<DI>()); // TWO
foo(new C<DDI>()); // THREE
foo(new C()); // FOUR
}
}
```

Which of the following options are correct?

a) Line marked with comment ONE will result in a compiler error.
b) Line marked with comment TWO will result in a compiler error.
c) Line marked with comment THREE will result in a compiler error.
d) Line marked with comment FOUR will result in a compiler error.

20. **Consider the following definitions:**
interface BI {}
interface DI extends BI {}
The following options provide definitions of a template class X. Which one of the options specifies class X with a type parameter whose upper bound declares DI to be the super type from which all type arguments must be derived?
a) class X <T super DI> { }
b) class X <T implements DI> { }
c) class X <T extends DI> { }
d) class X <T extends ? & DI> { }

Answers:

1.	9.	17.
2.	10.	18.
3.	11.	19.
4.	12.	20.
5.	13.	
6.	14.	
7.	15.	
8.	16.	