1 Consider the following code:

```
interface Greek { }

class Alpha implements Greek { }

class Beta extends Alpha {}

class Delta extends Beta {
public static void main( String[] args ) {
Beta x = new Beta();
// insert code here
}
}
```

Which of the following code snippet when inserted individual at the commented line (// insert code here), will cause a java.lang.ClassCastException?

Answer:

○ a. Greek f = (Beta)(Alpha)x;

○ b. Alpha a = x;

○ c. Greek f = (Alpha)x;

○ d. Beta b = (Beta)(Alpha)x;

● e. Greek f = (Delta)x;

2 Consider the following code snippet:

```
class Node {
Node node;
}

class NodeChain {
public static void main(String a[]) {
Node node1 = new Node(); // Line 1
node1.node = node1;
// Code here
}
}
```

Which of the following code snippets when replaced at the comment line (// Code Here) in the above code will make the object created at Line 1, eligible for garbage collection? (Choose 2)

Answer:
☐ a. node1.node = new Node();
☑ b. node1 = null;
☐ c. node1.node = null;
☐ d. node1 = node1.node;
☑ e. node1 = new Node();

3 You need to create a class that implements the Runnable interface to do background image processing. Out of the following list of method declarations, select the method that must satisfy the requirements.

Answer:
● a. public void run()
○ b. public void start()
○ c. public void stop()
○ d. public void suspend()

4 Consider the following code:

```
class Component {
private int param;
public void setParam(int param) { this.param = param; }
public int getParam() { return param; }
}

class Container {
private Component component;
public void setComponent(Component component) { this.component =
component; }
public Component getComponent() { return component; }
}

public class TestContainerComponent {
public static void main(String args[]) {
Container container = new Container();
Component component = new Component();

int j = 10;
component.setParam(j);
container.setComponent(component);

// Insert code here
System.out.println(container.getComponent().getParam());
}
}
```

Which of the following code snippets when individually repalced at the commented line (// Insert code here) in the above code will produce the output 100? (Choose 3)

Answer:

☐ a. container.setComponent(component); component=new Component(); component.setParam(100);

☑ b. component = new Component(); component.setParam(100); container.setComponent(component);

☑ c. component.setParam(100);

☑ d. component = new Component(); component.setParam(100);

☐ e. container.getComponent().setParam(100);

5 Consider the following code:

```
public class TestOne {
public static void main(String args[]) {
byte x = 3;
byte y = 5;
System.out.print((y%x) + ", ");
System.out.println(y == ((y/x) *x +(y%x)));
}
}
```

Which of the following gives the valid output for above?

Answer:
○ a. Prints: 1, true
● b. Prints 2, false
○ c. Prints: 1, false
○ d. Prints: 2, true

6 Consider the following code:

```
public class Code17 {
public static void main(String args[]) {
new Code17();
}
{
System.out.print("Planet ");
}
{
System.out.print("Welcome ");
}
}
```

Which of the following will be the valid output for the above code?

Answer: ○ a. Welcome Planet

○ b. Compilation Error

○ c. Compiles and Executes with no output

○ d. Planet

● e. Planet Welcome

7 Consider the following program:

```
class A implements Runnable {
public void run() {
System.out.print(Thread.currentThread().getName());
}
}

class B implements Runnable {
public void run() {
new A().run();
new Thread(new A(),"T2").run();
new Thread(new A(),"T3").start();
}
}

class C {
public static void main (String[] args) {
new Thread(new B(),"T1").start();
}
}
```

What will be the output of the above program?

Answer: ○ a. Prints: T1T2T2

○ b. Prints: T1T2T3

○ c. Prints: T1T1T2

● d. Prints: T1T1T3

○ e. Prints: T1T1T1

8 From JDK 1.6, which of the following interfaces is also implemented by java.util.TreeSet class?

Answer:

● a. NavigableSet

○ b. Deque

○ c. NavigableMap

○ d. NavigableList

9 Consider the following scenario:

Mr.Ram is working for a Software Company. He needs to save and reload objects from a Java application. For security reasons, the default Sun Microsystem's implementation of saving and loading of Java Objects is not used. He needs to write his own module for the same.

Which of the following options can be used to accomplish the above requirement?

Answer:

○ a. Cloneable interface

● b. Serializable interface

○ c. Readable interface

○ d. Externalizable interface

○ e. Writable interface

10

Consider the following code:

```
1. public class Garment {
2. public enum Color {
3. RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);
4. private final int rgb;
5. Color( int rgb) { this.rgb = rgb; }
6. public int getRGB() { return rgb; }
7. };
8. public static void main( String[] argv) {
9. // insert code here
10. }
11.}
```

Which of the following code snippets, when inserted independently at line 9, allow the Garment class to compile? (Choose 2)

Answer:

☑ a. Color treeColor = Color.GREEN;

☑ b. Color purple = new Color(0xff00ff);

□ c. Color skyColor = BLUE;

□ d. if(RED.getRGB() < BLUE.getRGB()) {}

□ e. if(Color.RED.ordinal() < Color.BLUE.ordinal()) {}

11

Consider the following code:

```
public abstract class Shape {
private int x;
private int y;

public abstract void draw();

public void setAnchor(int x, int y) {
this.x = x;
this.y = y;
```

}
}

Which of the following implementations use the Shape class correctly? (Choose 2)

Answer:

☑ a. public class Circle extends Shape {
private int radius;
public void setRadius(int radius) { this.radius = radius; }
public int getRadius() { return radius; }
public void draw() {/* code here */}
}

□ b. public class Circle extends Shape {
private int radius;
public void draw();
}

☑ c. public abstract class Circle extends Shape {
private int radius;
}

□ d. public class Circle implements Shape {
private int radius;
}

□ e. public class Circle extends Shape {
public int radius;
private void draw() {/* code here */}
}

12 Which of the following flow control features does Java support? (Choose 2)

Answer:

☑ a. Labeled continue

□ b. Labeled goto

□ c. Labeled throw

□ d. Labeled catch

☑ e. Labeled break

13 Consider the following code:

```
class Resource { }
class UserThread extends Thread {
public Resource res;
public void run() {
try {
synchronized(res) {
System.out.println("Planet");
res.wait();
Thread.sleep(1000);
res.notify();
System.out.println("Earth");
}
} catch(InterruptedException e) { }
}
}

public class StartUserThreads {
public static void main(String[] args) {
Resource r = new Resource();
UserThread ut1 = new UserThread();
ut1.res = r;
UserThread ut2 = new UserThread();
ut2.res = r;
ut1.start();
ut2.start();
}
}
```

Which of the following will give the correct output for the above code?

Answer: ○ a. Runtime Error "IllegalThreadStateException"

○ b. Prints the following output 2 times: Planet
(waits for 1000 milli seconds)
Earth

● c. Compile-time Error

○ d. Prints the following output 1 time: Planet
(waits for 1000 milli seconds)
Earth

○ e. Prints the following output:
Planet
Planet
(get into long wait. Never ends)

15 Consider the following code:

```
class Alpha {
protected Beta b;
}

class Gamma extends Alpha { }

class Beta { }
```

Which of the following statement is True?

Answer: ○ a. Beta has-a Gamma and Gamma is-a Alpha.

● b. Gamma has-a Beta and Gamma is-a Alpha

○ c. Alpha is-a Gamma and has-a Beta.

○ d. Gamma is-a Beta and has-a Alpha.

○ e. Alpha has-a Beta and Alpha is-a Gamma.

16 Consider the following scenario:

You are writing a set of classes related to cooking and have created your own exception hierarchy derived from java.lang.Exception as follows:

Exception
+-Bad TasteException
+-Bitter Exception
+-Sour Excpetion

BadTasteException is defined as an abstract class. You have a method eatMe that may throw a BitterException or a SourException.

Which of the following method declarations are acceptable to the compiler? (Choose 3)

Answer: ☑ a. public void eatMe() throws BitterException, SourException

□ b. public void eatMe() throws Throwable

□ c. public void eatMe() throw BadTasteException

☑ d. public void eatMe() throws BadTasteException

☑ e. public void eatMe() throws RuntimeException

17 Consider the following code snippet:

String deepak = "Did Deepak see bees? Deepak did.";

Which of the following method calls would refer to the letter b in the string referred by the variable deepak?

Answer: ○ a. charAt(13)

● b. charAt(15)

○ c. charAt(12)

○ d. charAt(16)

○ e. charAt(14)

18 Which of the following codes will compile and run properly?

Answer: ●

```
a. public class Test2 {
static public void main(String[] in) {
System.out.println("Test2");
}
}
```

○

```
b. public class Test4 {
static int main(String args[]) {
System.out.println("Test4");
}
}
```

○

```
c. public class Test1 {
public static void main() {
System.out.println("Test1");
}
}
```

○

```
d. public class Test3 {
public static void main(String args) {
System.out.println("Test3");
}
}
```

○

```
e. public class Test5 {
static void main(String[] data) {
System.out.println("Test5");
}
}
```

19 Consider the following code snippet:

```
import java.io.*;

public class IOCode1 {
public static void main(String args[]) throws IOException {
BufferedReader br1 = new BufferedReader(
new InputStreamReader(System.in));
BufferedWriter br2 = new BufferedWriter(
new OutputStreamWriter(System.out));

String line = null;
while( (line = br1.readLine()) != null ) {
br2.write(line);
br2.flush();
}
br1.close();
br2.close();
}
}
```

What will be the output for the above code snippet?

Answer:

○ a. Reads the text from keyboard and prints the same to the console on pressing Ctrl Z, flushes (erases) the same from the console.

○ b. Reads the text from keyboard character by character and prints the same to the console on typing every character.

● c. Reads the text from keyboard line by line and prints the same to the console on pressing ENTER key at the end of every line

○ d. Reads the text from keyboard line by line and prints the same to the console on pressing ENTER key at the end of every line, then the same is flushed (erased) from the console.

○ e. Reads the text from keyboard and prints the same to the console on pressing Ctrl Z

20 Which of the following options gives the relationship between a Pilot class and Plane class?

Answer:

○ a. Inheritance

○ b. Polymorphism

○ c. Persistence

○ d. Aggregation

● e. Association

21 Both TYPE_SCROLL_SENSITIVE and TYPE_SCROLL_INSENSITIVE types ResultSets will make changes visible if they are closed and then reopened. State True or False.

Answer: ○ True ● False

22 Consider the following code snippet:

```
import java.util.*;
import java.text.*;

public class TestCol5 {
public static void main(String[] args) {
String dob = "17/03/1981";
// Insert Code here
}
}
```

Which of the following code snippets, when substituted to (//Insert Code here) in the above program, will convert the dob from String to Date type?

Answer: ● a. DateFormat df = new SimpleDateFormat("dd/MM/yyyy");
try {

Date d = df.parse(dob);
} catch(ParseException pe) { }

○ b. GregorianCalendar g = new GregorianCalendar(dob);

Date d = g.getDate();

○ c. Calendar c = new Calendar(dob);
Date d = g.getDate();

○ d. Date d = new Date(dob);

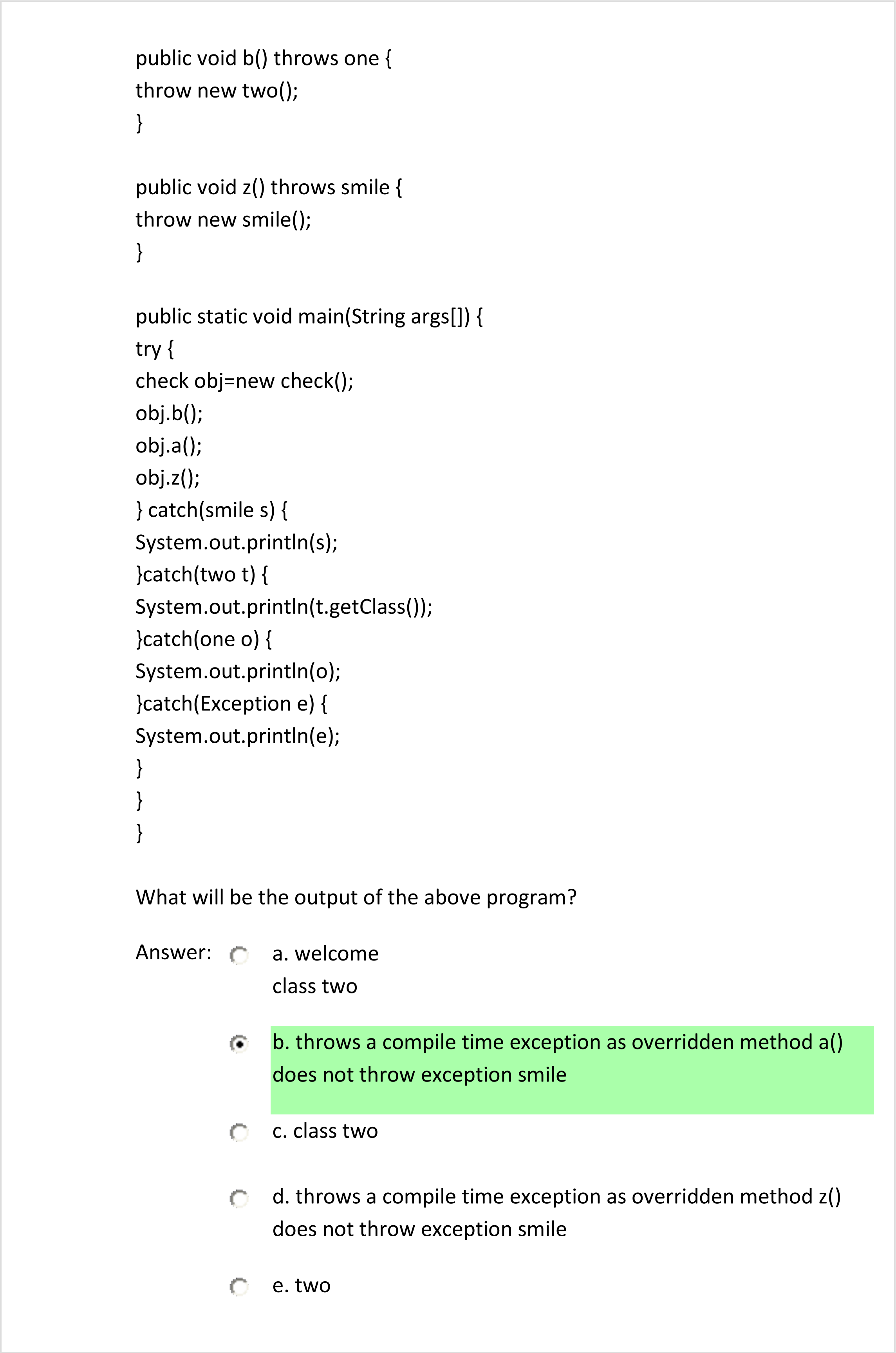
○ e. CalendarFormat cf = new SimpleCalendarFormat("dd/MM/yyyy");
try {
Date d = cf.parse(dob);
} catch(ParseException pe) { }

23 Consider the following program:

```
class joy extends Exception { }
class smile extends joy { }
interface happy {
void a() throws smile;
void z() throws smile;
}

class one extends Exception { }
class two extends one { }
abstract class test {
public void a()throws one { }
public void b() throws one { }
}

public class check extends test {
public void a() throws smile {
System.out.println("welcome");
throw new smile();
}
```



```
public void b() throws one {
throw new two();
}

public void z() throws smile {
throw new smile();
}

public static void main(String args[]) {
try {
check obj=new check();
obj.b();
obj.a();
obj.z();
} catch(smile s) {
System.out.println(s);
}catch(two t) {
System.out.println(t.getClass());
}catch(one o) {
System.out.println(o);
}catch(Exception e) {
System.out.println(e);
}
}
}
```

What will be the output of the above program?

Answer:

○ a. welcome
class two

● b. throws a compile time exception as overridden method a() does not throw exception smile

○ c. class two

○ d. throws a compile time exception as overridden method z() does not throw exception smile

○ e. two

24 Consider the following code:

```
public class WrapIt {
public static void main(String[] args) {
new WrapIt().testC('a');
}

public void testC(char ch) {
Integer ss = new Integer(ch);
Character cc = new Character(ch);
if(ss.equals(cc)) System.out.print("equals ");
if(ss.intValue()==cc.charValue()) {
System.out.println("EQ");
}
}
}
```

Which of the following gives the valid output for the above code?

Answer:

○ a. Compile-time error: Wrapper types cannot be compared using equals

● b. Compile-time error: Integer wrapper cannot accept char type

○ c. Prints: equals

○ d. Prints: EQ

○ e. Prints: equals EQ

25 Consider the following code snippet:

```
class TestString3 {
public static void main(String args[]) {
String s1 = "Hello";
StringBuffer sb = new StringBuffer(s1);
sb.reverse();
s1.concat(sb.toString());

System.out.println(s1 + sb + s1.length() + sb.length());
}
}
```

What will be the output of the above code snippet?

Answer:
● a. HelloolleH55

○ b. HelloolleH44

○ c. HelloHello44

○ d. HelloHello33

○ e. HelloHello55

26 The purpose of Soft Reference Type object is ______________.

Answer:
● a. to automatically delete objects from a container as soon clients stop referencing them

○ b. to allow clean up after finalization but before the space is reclaimed

○ c. to keep objects alive provided there is enough memory

○ d. to keep objects alive only while they are in use (reachable) by clients

27 Consider the following code:

```
public class Key1 {
public boolean testAns( String ans, int n ) {
boolean rslt;
if (ans.equalsIgnoreCase("YES") & n > 5)
rslt = true;

return rslt;
}

public static void main(String args[]) {
System.out.println(new Key1().testAns("no", 5));
}
}
```

Which of the following will be the output of the above program?

Answer:
○ a. Compile-time error
● b. Runtime Error
○ c. NO
○ d. false
○ e. true

28 Which of the following methods are defined in Object class? (Choose 3)

Answer:
☑ a. toString()
□ b. compareTo(Object)
☑ c. hashCode()
☑ d. equals(Object)
□ e. run()

29

Consider the following code:

```
public class Code4 {
private int second = first;
private int first = 1000;

public static void main(String args[]) {
System.out.println(new Code4().second);
}
}
```

Which of the following will be the output for the above code?

Answer:

○ a. Throws a Runtime error 'Illegal forward reference'

○ b. 1000

● c. Compiler complains about forward referencing of member variables first and second

○ d. Compiler complains about private memebers is not accessible from main() method

30

Which of the following are true about inheritance?(Choose 3)

Answer:

☑ a. Inheritance enables adding new features and functionality to an existing class without modifying the existing class

☑ b. In an inheritance hierarchy, a superclass can also act as a sub class

□ c. Inheritance is a kind of Encapsulation

□ d. Inheritance does not allow sharing data and methods among multiple classes

☑ e. In an inheritance hierarchy, a subclass can also act as a super class

31

Consider the following code snippet:

```
public class Welcome {
String title;
int value;

public Welcome() {
title += " Planet";
}

public void Welcome() {
System.out.println(title + " " + value);
}

public Welcome(int value) {
this.value = value;
title = "Welcome";
Welcome();
}

public static void main(String args[]) {
Welcome t = new Welcome(5);
}
}
```

Which of the following option will be the output for the above code snippet?

Answer:

○ a. Welcome Planet

○ b. The code runs with no output

○ c. Compilation fails

● d. Welcome 5

○ e. Welcome Planet 5

32

Which of the following are characteristics of Annotations?(Choose 2)

Answer: ☑ a. Annotations are not part of a program.

□ b. Annotations are predefined classes.

☑ c. Annotations are data about program.

□ d. Annotations are static methods.

33 Consider the following code:

```
public class ObParam{
public int b = 20;

public static void main(String argv[]){
ObParam o = new ObParam();
methodA(o);
}

public static void methodA(ObParam a) {
a.b++;
System.out.println(a.b);
methodB(a);
System.out.println(a.b);
}

public void methodB(ObParam b) {
b.b--;
}
}
```

Which of the following gives the correct output for the above code?

Answer: ○ a. Prints: 21 21

○ b. Prints: 20 21

● c. Compilation Error: Non-static method methodB() cannot be referenced from static context methodA()

○ d. Prints: 20 20

○ e. Prints: 21 20

34 Consider the following code snippet:

```
import java.util.*;

public class TestCol6 {
public static void main(String args[] ){
ArrayList a = new ArrayList(); // Line 1
a.add(new Integer(10));
a.add(new String("Hello"));
a.add(new Double(34.9));
}
}
```

Which of the following code snippets when replaced at the line marked //Line 1, will make the ArrayList a to accept only Wrapper types of primitive numerics?

Answer: ○ a. ArrayList<Double> a = new ArrayList<Double>();

○ b. ArrayList<Integer> a = new ArrayList<Double>();

○ c. ArrayList<WrapperType> a = new ArrayList<WrapperType>();

● d. ArrayList<Number> a = new ArrayList<Number>();

○ e. ArrayList<Integer> a = new ArrayList<Integer>();

35 Which of the following statements are true? (Choose 2)

Answer: □ a. All exceptions are thrown programmatically from the code or

API

□ b. All exceptions are thrown by JVM

□ c. JVM cannot throw user-defined exceptions

☑ d. JVM thrown exceptions can be thrown programmatically

☑ e. All RuntimeException are thrown by JVM

36 The following class definitions are in separate files. Note that the Widget and BigWidget classes are in different packages:

```
1. package conglomo;
2. public class Widget extends Object{
3. private int myWidth;
4. XXXXXX void setWidth( int n ) {
5. myWidth = n;
6. }
7. }

// the following is in a separate file and in separate package
8. package conglomo.widgets;
9. import conglomo.Widget ;
10. public class BigWidget extends Widget {
11. BigWidget() {
12. setWidth( 204 );
13. }
14. }
```

Which of the following modifiers, used in line 4 instead of XXXXXX, would allow the BigWidget class to access the setWidth method (as in line 12)? (Choose 2)

Answer: □ a. default (blank), that is, the method declaration would read void setWidth(int n)

□ b. final

☐ c. private

☑ d. protected

☑ e. public

37

Consider s1 and s2 are sets.

Which of the following options gives the exact meaning of the method call s1.retainAll(s2)?

Answer: ○ a. transforms s1 into the union of s1 and s2

○ b. transforms s1 into the (asymmetric) set difference of s1 and s2

● c. transforms s1 into the intersection of s1 and s2.

○ d. copies elements from s2 to s1

○ e. returns true if s2 is a subset of s1

38

Consider the following partial listing of the Widget class:

```
1. class Widget extends Thingee {
2. static private int widgetCount = 0;
3. static synchronized int addWidget() { widgetCount++;
4. return widgetCount;
5. }
6. String wName;
7. public Widget( int mx, String T ) {
8. wName = "I am Widget #" + addWidget();
9. }
10. // more methods follow
11. }
```

Which of the following gives the significance of the word "private" in line 2?

Answer: ● a. Since widgetCount is private, only methods in the Widget class can access it.

○ b. If another class tries to access widgetCount, a runtime exception will be thrown.

○ c. Since widgetCount is private, only the addWidget method can access it.

○ d. Since widgetCount is private, only methods in the Widget class and any derived classes can access it.

39 Consider the following program:

```
class CatchableException extends Throwable { }

class ThrowableException extends CatchableException { }

public class ThrowCatchable {
public static void main(String args[]) {
try {
tryThrowing();
}
catch(CatchableException c) {
System.out.println("Catchable caught");
}
finally {
tryCatching();
}
}

static void tryThrowing() throws CatchableException {
try {
tryCatching();
throw new ThrowableException();
}
catch(NullPointerException re) {
throw re;
}
}

static void tryCatching() {
System.out.println(null + " pointer exception");
}
}
```

What will be the output of the above program?

Answer:

○ a. null pointer exception
Catchable caught
null pointer exception

○ b. runtime error

● c. Catchable caught
null pointer exception
null pointer exception

○ d. compile-time error

○ e. null pointer exception
null pointer exception
Catchable caught

40 Which of the following statement is true?

Answer:

○ a. To call the join() method, a thread must own the lock of the object on which the call is to be made

○ b. To call the yield() method, a thread must own the lock of the object on which the call is to be made.

● c. To call the wait() method, a thread must own the lock of the object on which the call is to be made.

○ d. To call the wait() method, a thread must own the lock of the current thread.

○ e. To call the sleep() method, a thread must own the lock of the object which the call is to be made.

41 Which of the following methods is used to check whether ResultSet object contains records?

Answer:
○ a. last()
○ b. hasRecords()
○ c. previous()
● d. next()
○ e. first()

42 Which of the following are true about SQLWarning class in JDBC API?

Answer:
☑ a. SQLWarning is the subclass of SQLException
□ b. SQLWarning and SQLException can be used interchangeably
□ c. SQLWarning affects the normal program execution
□ d. SQLWarning messages are accessible through the Statement object
☑ e. SQLWarning can be caught and handled using try-catch

43 Consider the following code:

```
@interface Author {
String name();
String date();
```

}

Which of the following is the correct way of implementing the above declared annotation type?

Answer:

○ a. Author(
name = "Deepak",
date = "02/04/2008"
)
class MyClass() { }

○ b. @Author(
name = "Deepak",
)
class MyClass() { }

○ c. @Author(
"Deepak",
"02/04/2008"
)
class MyClass() { }

○ d. SELECT * BULK COLLECT INTO emp_details_t from emp;

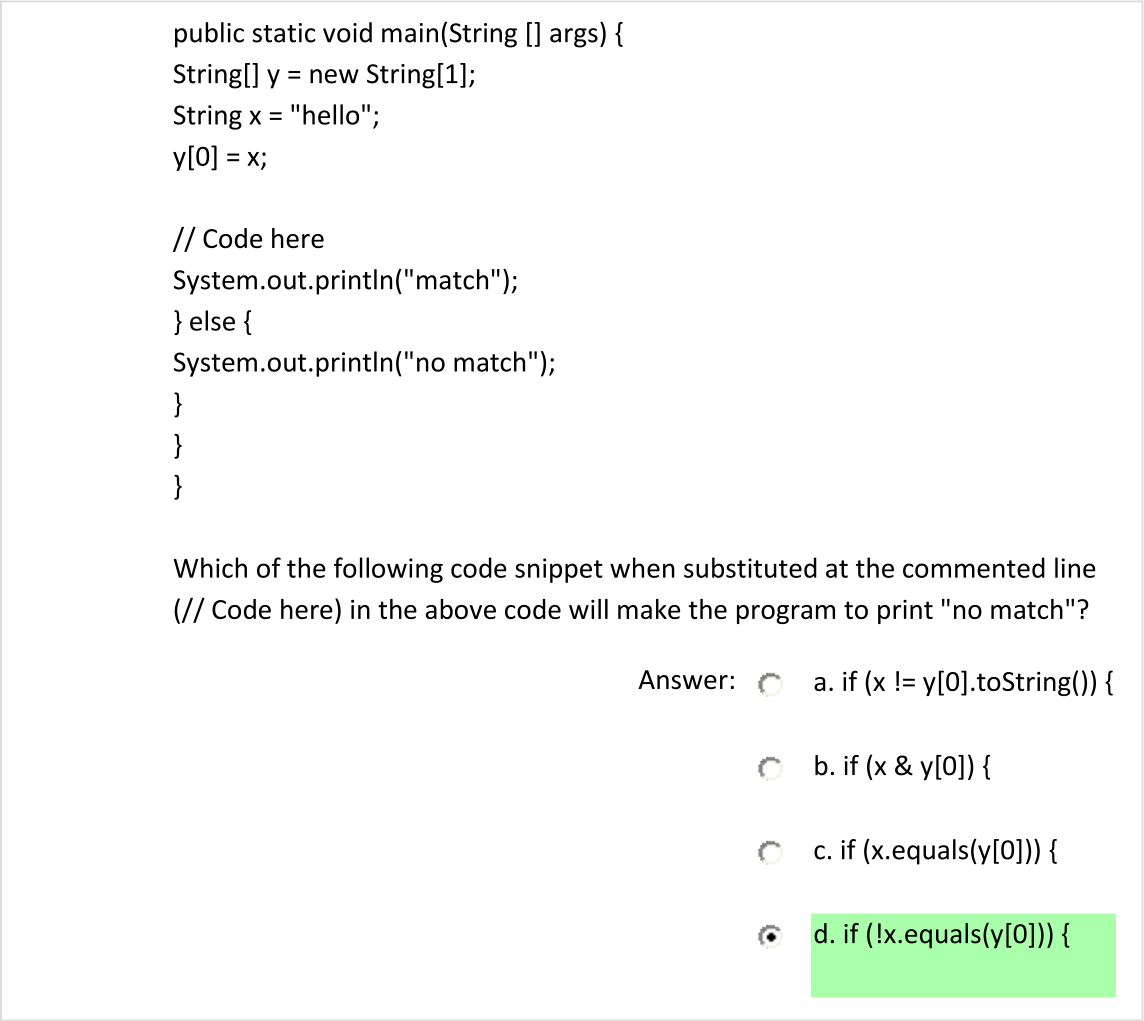
● e. @Author(
name = "Deepak",
date = "02/04/2008"
)
class MyClass() { }

44 Delimiters themselves be considered as tokens. State True or False.

Answer: ○ True ● False

45 Consider the following code:

public class ExampleSeven {



```
public static void main(String [] args) {
String[] y = new String[1];
String x = "hello";
y[0] = x;

// Code here
System.out.println("match");
} else {
System.out.println("no match");
}
}
}
```

Which of the following code snippet when substituted at the commented line (// Code here) in the above code will make the program to print "no match"?

Answer:
○ a. if (x != y[0].toString()) {

○ b. if (x & y[0]) {

○ c. if (x.equals(y[0])) {

● d. if (!x.equals(y[0])) {