Question 1

Assume that country is set for each class.
Given:
10. public class Money {
11. private String country, name;
12. public getCountry() { return country; }
13.}
and:
24. class Yen extends Money {
25. public String getCountry() { return super.country; }
26. }
27.
28. class Euro extends Money {
29. public String getCountry(String timeZone) {
30. return super.getCountry();
31. }
32. }
Which two are correct? (Choose two.)
A. Yen returns correct values.
B. Euro returns correct values.
C. An exception is thrown at runtime.
D. Yen and Euro both return correct values.
E. Compilation fails because of an error at line 25.
F. Compilation fails because of an error at line 30.

Answer: BE

Question 2

Given:
1. package sun.scjp;
2. public enum Color { RED, GREEN, BLUE }
1. package sun.beta;
2. // insert code here
3. public class Beta {
4. Color g = GREEN;
5. public static void main(String[] argv)
6. { System.out.println(GREEN); }
7. }
The class Beta and the enum Color are in different packages.
Which two code fragments, inserted individually at line 2 of the Beta declaration, will allow this code to compile? (Choose two.)
A. import sun.scjp.Color.*;

B. import static sun.scjp.Color.*;
C. import sun.scjp.Color; import static sun.scjp.Color.*;
D. import sun.scjp.*; import static sun.scjp.Color.*;
E. import sun.scjp.Color; import static sun.scjp.Color.GREEN;

Answer: CE

Question 3

A programmer is designing a class to encapsulate the information about an inventory item. A JavaBeans component is needed to do this. The InventoryItem class has private instance variables to store the item information:
10. private int itemId;
11. private String name;
12. private String description;
Which method signature follows the JavaBeans naming standards for modifying the itemId instance variable?
A. itemID(int itemId)
B. update(int itemId)
C. setItemId(int itemId)
D. mutateItemId(int itemId)
E. updateItemID(int itemId)

Answer: C

Question 4

Given:
12. public class AssertStuff {
13.
14. public static void main(String [] args) {
15. int x= *5;*
16. int y= 7;
17.
18. assert (x> y): “stuff”;
19. System.out.println(”passed”);
20. }
21. }
And these command line invocations:
java AssertStuff
java -ea AssertStuff
What is the result?

A. passed
stuff
B. stuff
passed
C. passed
An AssertionError is thrown with the word “stuff” added to the stack trace.
D. passed
An AssertionError is thrown without the word “stuff” added to the stack trace.
E. passed
An AssertionException is thrown with the word “stuff” added to the stack trace.
F. passed
An AssertionException is thrown without the word “stuff” added to the stack trace.

Answer: C

Question 5

Click the Exhibit button.

```
1. public class Test {
2.
3. public static void main(String [] args) {
4. boolean assert = true;
5. if(assert) {
6. System.out.println(”assert is true”);
7. }
8. }
9.
10. }
```

Given:
javac -source 1.3 Test.java
What is the result?
A. Compilation fails.
B. Compilation succeeds with errors.
C. Compilation succeeds with warnings.
D. Compilation succeeds without warnings or errors.

Answer: C

Question 6

Click the Exhibit button.
Class TestException
```
1. public class TestException extends Exception {
2. }
```
Class A:
```
1. public class A {
2.
3. public String sayHello(String name) throws TestException {
4.
5. if(name == null) {
6. throw new TestException();
7. }
8.
9. return "Hello "+ name;
10. }
11.
12. }
```
A programmer wants to use this code in an application:
45. A a=new A();
46. System.out.println(a.sayHello("John"));
Which two are true? (Choose two.)
A. Class A will not compile.
B. Line 46 can throw the unchecked exception TestException.
C. Line *45* can throw the unchecked exception TestException.
D. Line 46 will compile if the enclosing method throws a TestException.
E. Line 46 will compile if enclosed in a try block, where TestException is caught.

Answer: DE

Question 7

Given:
```
11.classA {
12. public void process() { System.out.print("A "); } }
13. class B extends A {
14. public void process() throws RuntimeException {
15. super.process();
16. if (true) throw new RuntimeException();
17. System.out.print("B"); }}
18. public static void main(String[] args) {
19. try { ((A)new B()).process(); }
20. catch (Exception e) { System.out.print("Exception "); }
21. }
```

What is the result?
A. Exception
B. A Exception
C. A Exception B
D. A B Exception
E. Compilation fails because of an error in line 14.
F. Compilation fails because of an error in line 19.

Answer: B

Question 8

Given:
11. public String makinStrings() {
12. String s = "Fred";
13. s = s + "47";
14. s = s.substring(2, 5);
15. s = s.toUpperCase();
16. return s.toString();
17. }
How many String objects will be created when this method is invoked?
A. 1
B. 2
C. 3
D. 4
E. 5
F. 6

Answer: C

Question 9

Given:
10. class MakeFile {
11. public static void main(String[] args) {
12. try {
13. File directory = new File("d");
14. File file = new File(directory,"f");
15. if(!file.exists()) {
16. file.createNewFile();
17. }
18. } catch (IOException e) {

19. e.printStackTrace
20. }
21. }
22. }
The current directory does NOT contain a directory named "d."
Which three are true? (Choose three.)
A. Line 16 is never executed.
B. An exception is thrown at runtime.
C. Line 13 creates a File object named "d."
D. Line 14 creates a File object named "f.'
E. Line 13 creates a directory named "d" in the file system.
F. Line 16 creates a directory named "d" and a file 'f' within it in the file system.
G. Line 14 creates a file named 'f' inside of the directory named "d" in the file system.

Answer: BCD

Question 10
Given:
10. public class Foo implements java.io.Serializable {
11. private int x;
12. public int getX() { return x; }
12.publicFoo(int x){this.x=x; }
13. private void writeObject(ObjectOutputStream s)
14. throws IOException {
15. // insert code here
16. }
17. }
Which code fragment, inserted at line *15,* will allow Foo objects to be correctly serialized and deserialized?
A. s.writeInt(x);
B. s.serialize(x);
C. s.writeObject(x);
D. s.defaultWriteObject();

Answer: D

Question 11

Given:
14. DateFormat df;
15. Date date = new Date();
16. //insert code here

17. String s = df.format(date);
Which two, inserted independently at line 16, allow the code to compile? (Choose two.)
A. df= new DateFormat();
B. df= Date.getFormatter();
C. df= date.getFormatter();
D. df= date.getDateFormatter();
E. df= Date.getDateFormatter();
F. df= DateFormat.getInstance();
G. df = DateFormat.getDateInstance();

Answer: FG

Question 12

Given:
11. String test = "Test A. Test B. Test C.";
12. // insert code here
13. String[] result = test.split(regex);
Which regular expression inserted at line 12 will correctly split test into "Test A," "Test B," and "Test C"?
A. String regex = "";
B. String regex = " ";
C. String regex = ".*".
D. String regex = "\\s"
E. String regex = "\\.\\s*";
F. String regex = "\\w[\.] +";

Answer: E

Question 13

Click the Exhibit button.
Given:
1. public class TwoThreads {
2
3. private static Object resource = new Object();
4.
5. private static void delay(long n) {
6. try { Thread.sleep(n); }
7. catch (Exception e) { System.out.print("Error "); }
8. }
9
10. public static void main(String[] args) {

```
11. System.out.print(”StartMain “);
12. new Thread1().start();
13. delay(1000);
14. Thread t2 = new Thread2();
15. t2.start();
16. delay(1000);
17. t2.interrupt
18. delay(1000);
19. System.out.print(”EndMain “);
20. }
21.
22. static class Thread 1 extends Thread {
23. public void run() {
24. synchronized (resource) {
25. System.out.print(”Startl “);
26. delay(6000);
27. System.out.print(”End1 “);
28. }
29. }
30. }
31.
32. static class Thread2 extends Thread {
33. public void run() {
34. synchronized (resource) {
35. System.out.print(”Start2 “);
36. delay(2000);
37. System.out.print(”End2 “);
38. }
39. }
40. }
41. }
```

Assume that sleep(n) executes in exactly m milliseconds, and all other code executes in an insignificant amount of time. What is the output if the main() method is run?

A. Compilation fails.
B. Deadlock occurs.
C. StartMain Start1 Error EndMain End1
D. StartMain Start1 EndMain End1 Start2 End2
E. StartMain Start1 Error Start2 EndMain End2 End1
F. StartMain Start1 Start2 Error End2 EndMain End1
G. StartMain Start1 EndMain End1 Start2 Error End2

Answer: G

Question 14

Click the Exhibit button.

```
10. public class Transfers {
11. public static void main(String[] args) throws Exception {
12. Record r1 = new Record();
13. Record r2 = new Record();
14. doTransfer(r1, r2, 5);
15. doTransfer(r2, r1, 2);
16. doTransfer(r1, r2, 1);
17. // print the result
18. System.out.println("rl = " + r1.get() +", r2=" + r2.get());
19. }
20. private static void doTransfer(
21. final Record a, final Record b, final int amount) {
22. Thread t = new Thread() {
23. public void run() {
24. new Clerk().transfer(a, b, amount);
25. }
26. };
27. t.start();
28. }
29. }
30. class Clerk {
31. public synchronized void transfer(Record a, Record b, int amount){
32. synchronized (a) {
33. synchronized (b) {
34. a.add(-amount);
35. b.add(amount);
36. }
37. }
38. }
39. }
40. class Record {
41.int num=10;
42. public int get() { return num; }
43. public void add(int n) { num = num + n; }
44. }
```

If Transfers.main() is run, which three are true? (Choose three.)

A. The output may be "r1 = 6, r2 = 14".

B. The output may be "r1 = *5,* r2 = *15".*

C. The output may be "r1 = 8, r2 = 12".

D. The code may run (and complete) with no output.

E. The code may deadlock (without completing) with no output.

F. M IllegalStateException or InterruptedException may be thrown at runtime.

Answer: ABE

Question 15

Click the Exhibit button.
1. public class Employee {
2. String name;
3. double baseSalary;
4. Employee(String name, double baseSalary) {
5. this.name = name;
6. this.baseSalary = baseSalary;
7. }
8. }
And:
1. public class Salesperson extends Employee {
2. double commission;
3. public Salesperson(String name, double baseSalary,
4. double commission) {
5. // insert code here
6. }
7. }
Which code, inserted at line 7, completes the Salesperson constructor?
A. this.commission = commission;
B. superb();
commission = commission;
C. this.commission = commission;
superb();
D. super(name, baseSalary);
this.commission = commission;
E. super();
this.commission = commission;
F. this.commission = commission;
super(name, baseSalary);

Answer: D

Question 16

Given:
1. public class Team extends java.util.LinkedList {
2. public void addPlayer(Player p) {

```
3. add(p);
4. }
5. public void compete(Team opponent) { /* more code here */ }
6. }
7. class Player { /* more code here */ }
```

Which two are true? (Choose two.)

A. This code will compile.
B. This code demonstrates proper design of an is-a relationship.
C. This code demonstrates proper design of a has-a relationship.
D. A Java programmer using the Team class could remove Player objects from a Team object.

Answer: AD

Question 17

Click the Exhibit button.

```
1. import java.util.*;
2. class KeyMaster {
3. public int i;
4. public KeyMaster(int i) { this.i = i; }
5. public boolean equals(Object o) { return i == ((KeyMaster)o).i; }
6. public int hashCode() { return i; }
7. }
8. public class MapIt {
9. public static void main(String[] args) {
10. Set<KeyMaster> set = new HashSet<KeyMaster>();
11. KeyMaster k1 = new KeyMaster(1);
12. KeyMaster k2 = new KeyMaster(2);
13. set.add(k1); set.add(k1);
14. set.add(k2); set.add(k2);
15. System.out.print(set.size() + ":");
16. k2.i = 1;
17. System.out.print(set.size() + ":");
18. set.remove(k1);
19. System.out.print(set.size() + ":");
20. set.remove(k2);
21. System.out.print(set.size());
22. }
23. }
```

What is the result?

A. 4:4:2:2
B. 4:4:3:2
C. 2:2:1:0

D. 2:2:0:0
E. 2:1:0:0
F. 2:2:1:1
G. 4:3:2:1

Answer: F

Question 18

Given:
1. import java.util.*;
2. public class Test {
3. public static void main(String[] args) {
4. List<String> strings = new ArrayList<String>();
5. // insert code here
6. }
7. }
Which four, inserted at line *5,* will allow compilation to succeed? (Choose four.)
A. String s = strings.get(0);
B. Iterator i1 = strings.iterator();
C. String[] array1 = strings.toArray();
D. Iterator<String> i2 = strings.iterator();
E. String[] array2 = strings.toArray(new String[1]);
F. Iterator<String> i3 = strings.iterator<String>();

Answer: ABDE

Question 19

Given:
classA {}
class B extends A {}
class C extends A {}
class D extends B {}
Which three statements are true? (Choose three.)
A. The type List<A> is assignable to List.
B. The type List<B> is assignable to List<A>.
C. The type List<Object> is assignable to List<?>.
D. The type List<D> is assignable to List<? extends B>.
E. The type List<? extends A> is assignable to List<A>.
F. The type List<Object> is assignable to any List reference.
G. The type List<? extends B> is assignable to List<? extends A>.

Answer: CDG

Question 20
Given:
11. public void addStrings(List list) {
12. list.add(”foo”);
13. list.add(”bar”);
14. }
What must you change in this method to compile without warnings?
A. add this code after line 11:
list = (List<String>) list;
B. change lines 12 and 13 to:
list.add<String>(”foo”);
list.add<String>(”bar”);
C. change the method signature on line 11 to:
public void addStrings(List<? extends String> list) {
D. change the method signature on line 11 to:
public void addStrings(List<? super String> list) {
E. No changes are necessary. This method compiles without warnings.

Answer: D

Question 21

Given:
1. public class Test {
2. public <T extends Comparable> T findLarger(T x, T y) {
3. if(x.compareTo(y) > 0) {
4. return x;
5. } else {
6. return y;
7. }
8. }
9. }
and:
22. Test t = new Test();
23. // insert code here
Which two will compile without errors when inserted at line 23?
(Choose two.)
A. Object x = t.findLarger(123, *”456”);*
B. int x = t.findLarger(123, new Double(456));
C. int x = t.findLarger(123, new Integer(456));
D. int x = (int) t.findLarger(new Double(123), new Double(456));

Answer: AC

Question 22

Given:
11. List list = // more code here
12. Collections.sort(list, new MyComparator());
Which code will sort this list in the opposite order of the sort in line 12?
A. Collections.reverseSort(list, new MyComparator());
B. Collections.sort(list, new MyComparator());
list.reverse();
C. Collections.sort(list, new InverseComparator(
new MyComparator()));
D. Collections.sort(list, Collections.reverseOrder(
new MyComparator()));

Answer: D

Question 23

Given:
ArrayList a = new ArrayList();
containing the values {"1", "2", "3", "4", "5", "6", "7", "8"}
Which code will return 2?
A. Collections. sort(a, a.reverse());
int result = Collections.binarySearch(a, "6");
B. Comparator c = Collections.reverseOrder();
Collections.sort(a, c);
int result = Collections.binarySearch(a, "6");
C. Comparator c = Collections.reverseOrder();
Collections.sort(a, c);
int result = Collections.binarySearch(a, "6",c);
D. Comparator c = Collections.reverseOrder(a);
Collections.sort(a, c);
int result = Collections.binarySearch(a, "6",c);
E. Comparator c = new InverseComparator(new Comparator());
Collections.sort(a);
int result = Collections.binarySearch(a, "6",c);

Answer: C

Question 24

Given a file GrizzlyBear.java:
1. package animals.mammals;
2.
3. public class GrizzlyBear extends Bear {
4. void hunt() {
5. Salmon s = findSalmon();
6. s.consume();
7. }
8. }
and another file, Salmon.java:
1. package animals.fish;
2.
3. public class Salmon extends Fish {
4. void consume() { /* do stuff */ }
5. }
Assume both classes are defined in the correct directories for theft packages, and that the Mammal class correctly defines the findSalmon() method. Which two changes allow this code to compile correctly? (Choose two.)
A. add public to the start of line 4 in Salmon.java
B. add public to the start of line 4 in GrizzlyBear.java
C. add import animals.mammals.*; at line 2 in Salmon.java
D. add import animals.fish.*; at line 2 in GrizzlyBear.java
E. add import animals.fish.Salmon.*; at line 2 in GrizzlyBear.java
F. add import animals.mammals.GrizzlyBear.*;at line 2 in Salmon.java

Answer: AD

Question 25

Given:
11. public class Counter {
12. public static void main(String[] args) {
13. int numArgs = /* insert code here *;*
14. }
15. }
and the command line:
java Counter one fred 42
Which code, inserted at line 13, captures the number of arguments passed into the program?
A. args.count
B. args.length

C. args.count()
D. args.length()
E. args.getLength()

Answer: B

Question 26

Click the Exhibit button.
11. class Payload {
12. private int weight;
13. public Payload(int wt) { weight = wt; }
13. public void setWeight(mt w) { weight = w; }
15. public String toString { return Integer.toString(weight); }
16. }
17.
18. public class TestPayload {
19. static void changePayload(Payload p) {
20. /* insert code here */
21. }
22.
23. public static void main(String[] args) {
24. Payload p = new Payload();
25. p.setWeight(1024);
26. changePayload(p);
27. System.out.println("The value of p is "+ p);
28. }
29. }
Which statement, placed at line 20, causes the code to print "The value of p is 420."?
A. p.setWeight(420);
B. p.changePayload(420);
C. p = new Payload(420);
D. Payload.setWeight(420);
E. p = Payload.setWeight(420);
F. p = new Payload();
p.setWeight(420);

Answer: A

Question 27

Given:
42. public class ClassA {

43. public int getValue() {
44.int value=0;
45. boolean setting = true;
46. String title=”Hello”;
47. if (value || (setting && title == “Hello”)) { return 1; }
48. if (value == 1 & title.equals(”Hello”)) { return 2; }
49. }
50. }
And:
70. ClassA a = new ClassA();
71. a.getValue();
What is the result?
A. 1
B. 2
C. Compilation fails.
D. The code runs with no output.
E. An exception is thrown at runtime.

Answer: C