For questions 1 – 3, use the following partial class definitions:

```
public class A1 {
   public int x;
   private int y;
   protected int z;
   …
}
public class A2 extends A1 {
   protected int a;
   private int b;
   …
}
public class A3 extends A2 {
   private int q;
   …
}
```

1) Which of the following lists of instance data are accessible in class A2?
   a) `x, y, z, a, b`
   b) `x, y, z, a`
   c) `x, z, a, b`
   d) `z, a, b`
   e) `a, b`

2) Which of the following lists of instance data are accessible in A3?
   a) `x, y, z, a, b, q`
   b) `a, b, q`
   c) `a, q`
   d) `x, z, a, q`
   e) `x, a, q`

3) Which of the following is true regarding the use of instance data y of class A1?
   a) it is accessible in A1, A2 and A3
   b) it is accessible in A1 and A2
   c) it is accessible only in A1
   d) it is accessible only in A3
   e) it is not accessible to any of the three classes

4) The instruction super( ); does which of the following?
   a) calls the method super as defined in the current class
   b) calls the method super as defined in the current class' parent class
   c) calls the method super as defined in java.lang
   d) calls the constructor as defined in the current class
   e) calls the constructor as defined in the current class' parent class

5) Aside from permitting inheritance, the visibility modifier *protected* is also used to
   a) permit access to the protected item by any class defined in the same package
   b) permit access to the protected item by any static class
   c) permit access to the protected item by any parent class

d) ensure that the class can not throw a NullPointerException
e) define abstract elements of an interface

6) All classes in Java are directly or indirectly subclasses of the _______ class.
a) Wrapper
b) String
c) Reference
d) this
e) Object

For questions 7 – 9, assume that Student, Employee and Retired are all extended classes of Person, and all four classes have different implementations of the method getMoney. Consider the following code where … are the required parameters for the constructors:

```
Person p = new Person(…);
int m1 = p.getMoney( );             // assignment 1
p = new Student(…);
int m2 = p.getMoney( );             // assignment 2
if (m2 < 100000)
   p = new Employee(…);
else if (m1 > 50000)
   p = new Retired(…);
int m3 = p.getMoney( );             // assignment 3
```

7) The reference to `getMoney()` in assignment 1 is to the class
a) Person
b) Student
c) Employee
d) Retired
e) none of the above, this cannot be determined by examining the code

8) The reference to `getMoney()` in assignment 2 is to the class
a) Person
b) Student
c) Employee
d) Retired
e) none of the above, this cannot be determined by examining the code

9) The reference to `getMoney()` in assignment 3 is to the class
a) Person
b) Student
c) Employee
d) Retired
e) none of the above, this cannot be determined by examining the code

10) The relationship between a child (sub) class and a parent (super) class is referred to as a(n) ____ relationship.
a) has-a
b) is-a
c) was-a
d) instance-of

For questions 11 – 12, consider the following class definition:

```
public class Test {
   private int x;

   public test (int newValue) {
      x = newValue;
   }
}
```

11) Which of the following is true about the class `Test`?

a) it has no parent class
b) it's parent class is `Object`
c) it's parent class is `Java`
d) it can not be extended
e) it has a default child called `Object`

12) If q1 and q2 are objects of Test class, then `q1.equals(q2)`

a) is a syntax error since equals is not defined in the test class
b) is true if q1 and q2 both store the same value of x
c) is true if q1 and q2 reference the same test object
d) is never true
e) throws a NullPointerException

13) What does the following code do? Assume list is an initialized array of int values, temp is some previously initialized int value, and c is an int initialized to 0.

```
for (int j = 0; j < list.length; j++)
   if (list[j] < temp)
      c++;
```

a) It finds the smallest value and stores it in temp
b) It finds the largest value and stores it in temp
c) It counts the number of elements equal to the smallest value in list
d) It counts the number of elements in list that are less than temp
e) It sorts the values in list to be in ascending order

Use the code below to answer questions 14 – 16. Note that the catch statements in the code are not implemented, but you will not need those details. Assume filename is a String, x is an int, a is a double array and I is an int. Use the comments `I1, I2, I3, e1, e2, e3, e4, e5` to answer the questions (I for instruction, e for exception handler).

```
try {
   BufferedReader infile = new BufferedReader(new FileReader(filename)); //i1
   int x = Integer.parseInt(infile.readLine( )); //i2
    a[++i] = (double) (1 / x);   //i3
}
catch (FileNotFoundException ex) {…}        // e1
catch (NumberFormatException ex) {…}         // e2
catch (ArithmeticException ex) {…}          // e3
catch (ArrayIndexOutOfBounds ex) {…}         // e4
catch (IOException ex) {…}                    // e5
```

14) An exception raised by the instruction in i1 would be caught by the catch statement labeled

a) `e1`
b) `e2`
c) `e5`
d) `either e1 or e5`
e) `either e1, e4 or e5`

15) An exception raised by the instruction in i2 would be caught by the catch statement labeled

a) `e1`
b) `e2`
c) `e3`
d) `e5`
e) `either e2 or e5`

16) An exception raised by the instruction in i3 would be caught by the catch statement labeled

a) `e2`
b) `e3`
c) `e4`
d) `either e3 or e4`
e) `either e2, e3 or e4`

17) Consider the array declaration and instantiation: int[ ] arr = new int[5]; Which of the following is true about arr?

a) It stores 5 elements with legal indices between 1 and 5
b) It stores 5 elements with legal indices between 0 and 4
c) It stores 4 elements with legal indices between 1 and 4
d) It stores 6 elements with legal indices between 0 and 5
e) It stores 5 elements with legal indices between 0 and 5

18) Which of the following messages passed to the String str could throw a StringIndexOutOfBoundsException?

a) `str.length()`
b) `str.charAt(2);`
c) `str.replace('a', 'A');`
d) `str.equals(str);`
e) any of the above could throw a StringIndexOutOfBoundsException

19) Assume Exceptionname is a checked exception. If a method uses a class that can generate *Exceptionname*, then either the method must include try and catch statements where a catch statement catches *Exceptionname*, or the method header must include the statement

a) throw *Exceptionname*
b) throws *Exceptioname*
c) catch *Exceptionname*
d) catches *Exceptionname*
e) implements *Exceptionname*

20) assume x and y are String variables with x = "Hello" and y = null.
If the operation `y = "Hello";` is performed, then the result of `(x == y)` is

a) true

b) false
c) x and y becoming aliases
d) x being set to the value null
e) a run-time error

21) Assume infile is a BufferedReader for a textfile and that the textfile is empty. What is returned from the message infile.readLine( ); ?

a) 0
b) null
c) a special character known as the End-of-file marker (EOF)
d) none of the above, the message causes a NullPointerException to be thrown
e) none of the above, the message causes a EndOfFileException to be thrown

22) When a program terminates because a thrown exception is not handled, the program

a) starts up again automatically
b) opens a dialog box, which asks the user whether the program should start again, end, or enter a debugging mode
c) saves all output to a disk file called the "runStackTrace.txt"
d) opens a dialog box for the user to specify a disk file name, and all output is stored to that disk file
e) outputs a message indicating what and where the exception was thrown

For questions 23-26 assume values is an int array that is currently filled to capacity, with the following values:

9	4	12	2	6	8	18

23) What is returned by values[1]?
24) What is the value of values.length?

25) Which of the following loops would adequately add 1 to each element stored in values?

```
a) for (int j=1; j < values.length ;  j++)  values[j]++;
b) for (int j=0; j < values.length ;  j++)  values[j]++;
c) for (int j=0; j <= values.length;  j++)  values[j]++;
d) for (int j=0; j < values.length-1;  j++) values[j]++;
e) for (int j=1; j < values.length-1;  j++)  values[j]++;
```

26) The statement `System.out.println(values[7]);` will

a) output 7
b) output 18
c) output nothing
d) cause an ArrayOutOfBoundsException to be thrown
e) cause a syntax error

For questions 27-29 use the following class.

```
class Node {
   int  info;
   Node next;
}
```

27) Assume Node temp references the last element of the linked list. Which of the following conditions is true about temp?

```
a)  (temp.info == 0)
b)  (temp.next == null)
c)  (temp == head)
d)  (temp == null)
e)  (temp.next == null && temp.info == null)
```

28) Assume that the linked list has at least two Nodes in it. Which of the following instructions will return the second int value in the list?

```
a)  return head.info;
b)  return head.next.info;
c)  return head.next.next.info;
d)  return head.next.next.next.info;
```

e) It is not possible to return the second int value in the list using head.

29) Assume Node temp is currently set equal to head. Which of the following while loops is the best to be used to iterate through each element of a linked list?

```
a)  while (head != null)
       head = temp.next;
b)  while (temp != null)
       temp = temp.next;
c)  while (head != null)
       temp = temp.next;
d)  while (head != null)
       head = head.next;
e)  while (temp != null)
       head = head.next;
```

For questions 30 – 31, consider the following class definition:

```
public class Aclass {
   private int x;
   protected int y;

   public Aclass (int a, int b) {
      x = a;
      y = b;
   }

   public int addEm () {
      return x + y;
   }

   public String toString () {
      return “” + x + “ ” + y;
   }
}
```

30) Consider that you want to extend AClass to BClass. BClass will have a third int instance data, z. Which of the following would best define BClass' constructor?

```
a) public BClass (int a, int b, int c) {
      super(a, b, c);
   }

b) public BClass (int a, int b, int c) {
      x = a;
      y = b;
      z = c;
   }
c) public BClass (int a, int b, int c) {
      z = c;
   }

d) public BClass (int a, int b, int c) {
      super(a, b);
      z = c;
   }
e) public BClass (int a, int b, int c) {
      super( );
   }
```

31) Which of the following would best redefine the toString method for BClass?

```
a) public String toString () {
      return super.toString();
   }

b) public String toString () {
      return super.toString() + “ ” + z;
   }

c) public String toString () {
      return super.toString() + “ ” + x + “ ” + y + “ ” + z;
   }

d) public String toString () {
      return “” + x + “ ” + y + “ ” + z;
   }
```