INSTRUCTIONS TO CANDIDATES

NAME OF CANDIDATE .................................................. SEAT NO ..........................

USUAL SIGNATURE  ..................................................

READ THE FOLLOWING CAREFULLY:

1. Each of the following questions comprise 5 statements, for which you should select the one most appropriate answer.

2. On the question paper, place a tick in the box to indicate your answer.

3. Enter your name and examination number IN PENCIL on the computer answer sheet according to the instruction on that sheet. Please note that all numbers MUST consist of three digits, e.g. 9 is entered as 009 and 25 as 025. The digits should be entered in the boxes under ‘Candidate Number’ and entered by means of horizontal lines in the appropriate boxes underneath, exactly as when answering questions.

4. When you have completed this question paper, read the instructions on the computer answer sheet carefully and transfer your answers. Use a HB pencil to mark the computer answer sheet and if you change your mind be sure to erase the mark you have made. You may then mark the alternative answer or mark the answer as don’t know.

5. At the end of the examination, be absolutely sure to hand in BOTH the answer sheet bearing the questions AND the computer answer sheet.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM
Questions 1 and 2 refer to the following Java declarations.

```java
class APrinter
{
    APrinter() { System.out.print("A"); }
    APrinter(int i) { System.out.print("A"+i); }
}

class BPrinter extends APrinter
{
    BPrinter() { System.out.print("B"); }
    BPrinter(int i)
    {
        super(i);
        System.out.print("B");
    }
}

class CPrinter extends BPrinter
{
    CPrinter(int i) { System.out.print("C"+i); }
}
```

1. Consider the Java declarations above. What would you expect to see on standard output when an instance of BPrinter is created with `new BPrinter(1)`?
   - A. AA1B
   - B. A1B
   - C. AB1B
   - D. B1B
   - E. BB

2. Look again at the Java code above. What would you expect to see on standard output when an instance of CPrinter is created with `new CPrinter(2)`?
   - A. C2
   - B. BC2
   - C. B2C2
   - D. ABC2
   - E. A2BC2
3. Suppose the following two classes are declared in separate files in the same directory:

class Point
{
    int xCoord, yCoord;

    Point(int x, int y) { xCoord = x; yCoord = y; }

    private void scale(int factor)
    {
        xCoord *= factor; yCoord *= factor;
    }
}

class ThreeDPoint extends Point
{
    int zCoord;

    ThreeDPoint(int x, int y, int z)
    {
        xCoord = x; yCoord = y; zCoord = z;
    }

    private void scale(int factor)
    {
        super.scale(factor); zCoord *= factor;
    }
}

Possible causes for compile-time errors might be:

1. the class Point is not public, and therefore cannot be subclassed by ThreeDPoint;
2. the fields xCoord and yCoord are not protected, and therefore not in scope in class ThreeDPoint;
3. class Point does not have a constructor with no arguments;
4. method scale(int) is private, and not in scope in class ThreeDPoint;
5. the keyword super can only be used to refer to constructors.

Which of these are real causes of compile-time errors?

A. 1 and 5 only.    
B. 2 and 4 only.    
C. 3 and 4 only.    
D. 2, 3 and 4 only.  
E. 2, 4 and 5 only.
4. For a given class and abstract data type, which of the following best describes the meaning of ‘adequacy of representation’?

A. The class has no public constructor, but provides public methods that create instances of the class.
B. The class has a field that stores values of the abstract data type.
C. Every value of the abstract data type is represented by an instance of the class, and every instance of the class represents a data value.
D. Every operation in the abstract data type is implemented by a method in the class.
E. Each public method in the class implements one of the operations in the abstract data type.

5. Which of the following best describes polymorphic code in Java?

A. Polymorphic code treats a variety of related classes in a generic way.
B. Polymorphic code uses the `instanceof` operator to determine the type of an instance.
C. Polymorphic code stores instances of different types in one variable.
D. Polymorphic code uses casts to ensure that all instances belong to the correct class.
E. Polymorphic code works with a variety of classes by providing methods for each class.

6. Consider the following statements about javadoc comments.

1. An asterisk at the start of a line in a javadoc comment will be ignored.
2. The first sentence of a javadoc comment should be a brief summary of the class or member being described.
3. The first line of a javadoc comment should contain only the characters ‘//**’.
4. A javadoc comment can contain HTML tags.

Which of these statements are true?

A. 1 and 2 only.
B. 1 and 3 only.
C. 1, 2 and 4 only.
D. 2, 3 and 4 only.
E. All of them.

7. Suppose a class `Inner` is declared within a class `Outer`; which of the following statements is false?

A. `Inner` can be declared `public, private, or protected`.
B. All members of `Outer` are visible within `Inner`.
C. Private methods of `Inner` are not visible in any other classes declared within `Outer`.
D. `Inner` must be declared `static`.
E. `static` members can be declared within `Inner`.
8. Consider the following statements about anonymous classes.

1. An anonymous class is always an implementation of an interface.
2. An anonymous class is always an inner class.
3. An anonymous class can be a subclass of an abstract class.
4. An anonymous class can be a subclass of another anonymous class.

Which of these statements are true?

A. 2 and 4 only.                         ☐
B. 2 and 3 only.                         ☐
C. 1, 2 and 4 only.                      ☐
D. 1, 2 and 3 only.                      ☐
E. All of them.                          ☐

9. Consider the following code:

```java
class Test {
    public interface Inner {
        public int getValue();
    }

    public Inner makeInner(int i) {
        return new Inner()
            { public int getValue() { return i; } };
    }
}
```

This will raise a compile-time error; why?

A. Interfaces cannot be declared within a class. ☐
B. The variable i should be declared final. ☐
C. The variable i should be declared static. ☐
D. There should not be a semicolon after getValue(). ☐
E. There should not be a semicolon on the second-last line. ☐
10. Suppose the following declarations are contained in a file Test.java:

```java
interface Number
{
    public int getValue();
}

class Test
{
    class Zero implements Number
    {
        public int getValue()
        {
            return 0;
        }
    }

    public static final Number zero = new Zero();
}
```

This will raise a compile-time error; why?

A. Number is not public and therefore not in scope within Test.  
B. Zero is not public and therefore an instance of Zero cannot be assigned to the public field zero.  
C. Objects cannot be stored as final constants.  
D. The field zero is declared final, so the class Zero should be declared final.  
E. The field zero is declared static, but the class Zero is not declared static.

11. When an exception is thrown, which of the following, in general, is the best course of action?

A. Always print out the entire stack-trace, so that the user has as much information as possible about the error.  
B. Try to catch the exception as high up in the stack trace (closest to the error) as possible, because this is where most information about the cause of the error is available.  
C. Try to catch the exception as deep down in the stack trace (closest to the top level) as possible, because this is where the most informative error message can be formulated.  
D. Never try to catch the error, because the Java interpreter does that anyway.  
E. Never report an error to the user, because that will only diminish their faith in the program.
12. Consider the following declarations.

```java
static void printArray(int[] a, int i)
{
    System.out.println(a[i]); // print the ith element
    printArray(a,i+1); // repeat for i+1
}

public static void main(String[] args)
{
    int[] is = new int[]{1,2,3,4};
    printArray(is,0);
}
```

When the `main()` method is run, an exception is thrown; how many method calls are in the stack trace?

A. One.  
B. Two.  
C. Four.  
D. Six.  
E. Eight.

13. Suppose you are writing a parser for a programming language as part of a window-based program editor and compiler. You might decide to use a class `ParseException` to handle the case where the input program is not syntactically well-formed. Colleagues propose the following courses of action:

1. `ParseException` should extend `Exception` so that instances can be thrown with the keyword `throw`;
2. `ParseException` should extend `RuntimeException` because instances would only be thrown at run-time;
3. `ParseException` should extend `RuntimeException` so that methods do not need to advertise that they might throw `ParseException`;
4. all methods should advertise rather than catch `ParseException` so that syntactic errors can be reported by the top-level program editor.

Which of these suggestions should you follow?

A. 4 only.  
B. 1 and 4 only.  
C. 2 and 3 only.  
D. 1, 3 and 4 only.  
E. 2, 3 and 4 only.
14. Which of the following is a true statement about time-slicing in Java?
   A. Time-slicing requires a multiple-processor architecture.  
   B. Time-slicing can only be ensured if every thread calls the sleep() method.  
   C. Time-slicing can only be ensured if every thread calls the wait() and notify() methods.  
   D. In time-slicing, a thread is allowed to run for a period of time that is specified by the programmer.  
   E. In time-slicing, a thread is allowed to run for a period of time that is determined by the interpreter.  

15. Which of the following best describes the effect of the Thread.sleep() method in a multithreaded program?
   A. Calling sleep() causes execution of the thread to pause until another thread calls sleep().  
   B. Calling sleep() causes execution of the thread to pause until another thread calls the wake() method.  
   C. Calling sleep() causes execution of the thread to end.  
   D. Calling sleep() causes execution of a thread to pause for a specified amount of time, after which the thread is ready to run.  
   E. Calling sleep() causes execution of a thread to pause for a specified amount of time, after which execution of the thread immediately resumes.  

16. In the context of multithreaded programs, what is ‘interference’?
   A. Interference is when one thread calls the interrupt() method of another thread.  
   B. Interference is when one thread prevents another thread from accessing a shared resource.  
   C. Interference is the process of stopping one thread running and starting another.  
   D. Interference is a form of deadlock caused by two threads reading the value of a shared variable at the same time.  
   E. Interference is data corruption that arises when two or more threads access a shared resource.  

17. Thread-safety can be ensured in Java by qualifying a method with the keyword synchronized. This ensures that synchronized methods cannot be called concurrently in different threads. Notionally, the interpreter associates a ‘key’ with the synchronized method, and any object that calls the synchronized method must first obtain the key. Concerning the number of keys, which of the following is true?
   A. Each instance of the class has one key for each synchronized method in the class.  
   B. Each instance of the class has one key, regardless of the number of synchronized methods in the class.  
   C. Each instance of each class that can call the synchronized method has its own key.  
   D. There is one key for each synchronized method in the class, and these keys are shared by all instances of the class.  
   E. There is one key, shared by all instances of the class, regardless of the number of instances of the class.
18. Consider the following fragment of code:

```java
synchronized(new Integer(0))
{
    for (int i=0; i<5; i++)
    {
        System.out.println();
    }
}
```

Which of the following best describes the correctness of the code?

A. The code will compile, but it will probably not behave as expected, because it will prevent all other classes from referring to `System.out` at run-time.  

B. The code will compile, but it will probably not behave as expected, because `new Integer(0)` will create a new key-keeper each time the code is executed.  

C. The code will compile, but it will almost certainly cause deadlock.  

D. The code will cause a compile-time error because `new Integer(0)` is not static.  

E. The code will cause a compile-time error because `System.out` is not static.  

19. Consider the following statements about the `Thread.wait()` method.

1. When `wait()` is invoked in a thread, that thread is put in the pool of waiting threads, keeping the monitor key for the synchronized method.

2. When `wait()` is invoked in a thread, that thread is put in the pool of waiting threads, and releases the monitor key for the synchronized method; when the thread is next invoked, any other thread that has the monitor key must release it and give it back to the thread.

3. When `wait()` is invoked in a thread, that thread is put in the pool of waiting threads, and releases the monitor key for the synchronized method; when the thread is next invoked, it must reacquire the monitor key before it can proceed.

4. For each call of `wait()` there should be a matching call of either `notify()` or `notifyAll()`.

5. Using the `wait()` method can allow more than one thread to access a critical section at the same time.

Which of these statements are true?

A. 1 and 4 only.  

B. 2 and 4 only.  

C. 3 and 5 only.  

D. 1, 4 and 5 only.  

E. 3, 4 and 5 only.
20. Which of the following best describes the effect of calling the method `notify()` within a synchronized method?

A. When `notify()` is called, all threads in the waiting pool for the monitor key are moved to the ready-pool. □
B. When `notify()` is called, one thread in the waiting pool for the monitor key is chosen to be moved to the ready-pool. □
C. When `notify()` is called, the thread that currently holds the monitor key is moved to the ready-pool. □
D. When `notify()` is called, the thread that currently holds the monitor key is chosen to execute. □
E. When `notify()` is called, the interpreter shuts down the monitor on the synchronized method. □

21. Consider the following statements concerning threads.

1. Threads should always be stopped by calling the `Thread.stop()` method.
2. Threads should always be started by calling the `Thread.run()` method.
3. Threads should always be started by calling the `Thread.start()` method.
4. Threads may be started by calling the `Thread(Runnable)` constructor.
5. Threads should never be created by subclassing `Thread` and calling the `Thread()` constructor.

Which of these statements are true?

A. 1 only. □
B. 3 only. □
C. 1 and 2 only. □
D. 3 and 4 only. □
E. 3 and 5 only. □

22. Consider the following possible features of a user interface:

1. Results of actions are immediately visible.
2. Interaction takes place through a touch-sensitive screen.
3. Objects in the application domain are represented visually.
4. Actions are reversible.

Which of these features are characteristic of direct manipulation?

A. 1, 2 and 3 only. □
B. 1, 3 and 4 only. □
C. 1 and 3 only. □
D. 2 and 3 only. □
E. 2 and 4 only. □
23. Which of the following describes ‘Knowledgeable, Intermittent Users’?

A. High levels of both syntactic and semantic knowledge. ☐
B. Limited syntactic knowledge, but a high level of semantic knowledge. ☐
C. A high level of syntactic knowledge, but limited semantic knowledge. ☐
D. A high level of syntactic knowledge, but limited knowledge of task concepts. ☐
E. High levels of both semantic knowledge and knowledge of task concepts. ☐

24. Consider the following features of user interfaces:

1. use of a command language
2. interaction through form fill-in
3. support for macro definition
4. informative feedback provided on each task accomplished.

Which of these are particularly suited to expert users?

A. 1 only. ☐
B. 1 and 3 only. ☐
C. 2 and 3 only. ☐
D. 1, 3 and 4 only. ☐
E. 2, 3 and 4 only. ☐

25. Which of the following best describes the Model-View-Controller architecture?

A. Data, display, and user controls are implemented in separate packages. ☐
B. Data, display, and user controls are implemented in separate classes. ☐
C. Data is implemented in a separate package, and there is a clear separation between the implementation of functionalities for display and data manipulation. ☐
D. Data is implemented in one class, and there is a clear separation between the implementation of functionalities for display and data manipulation. ☐
E. Data, display, and user controls are implemented in separate threads. ☐

26. Which of the following best describes how AWT events are handled in Java?

A. Each AWT event causes a new thread to be created to handle that event. ☐
B. All AWT events are handled in the EventDispatch thread. ☐
C. All AWT events are handled in the main thread. ☐
D. The programmer must explicitly create one thread to handle all AWT events. ☐
E. For each AWT event, the programmer must explicitly create a thread to handle that event. ☐
27. Which of the following is the most accurate statement about thread-safety for graphical user-interfaces in Java?

A. An interface is guaranteed to be thread-safe if it uses only javax.swing components. 

B. An interface is guaranteed to be thread-safe if it uses only java.awt components. 

C. An interface is guaranteed to be thread-safe if it uses only the SwingEvent thread. 

D. An interface is guaranteed to be thread-safe if it uses only the EventDispatch thread. 

E. An interface is guaranteed to be thread-safe if a separate thread is created for each event. 

28. Below are some subclasses of java.awt.AWTEvent:

1. KeyEvent 
2. TextEvent 
3. MouseEvent 
4. ActionEvent 

Which of these can be handled by a listener registered with a TextField component?

A. 1 and 2 only. 

B. 2 and 3 only. 

C. 1, 2 and 4 only. 

D. 2, 3 and 4 only 

E. All of them. 

29. Which of the following best describes how component-generated events can be handled in Java?

A. All component-generated events are handled automatically by the Java interpreter. 

B. The programmer creates instances of appropriate listener classes, and registers these instances with the component; when the component generates an event, the relevant methods of all registered listeners are executed. 

C. The programmer creates instances of appropriate listener classes, and registers these instances with the component; when the component generates an event, one of the registered listeners is chosen by the interpreter and the relevant method of that instance is executed. 

D. The programmer must declare a subclass of the component and add one method for each type of event the component can generate; when the component generates an event, the appropriate method is executed. 

E. Each component class has event-handling methods; the programmer declares a subclass of the component and overrides those methods; when the component generates an event, the overridden method is executed.
30. What is the immediate superclass of `javax.swing.JFrame`?

   A. `java.lang.Object`
   B. `javax.swing.Window`
   C. `javax.swing.JComponent`
   D. `java.AWT.Window`
   E. `java.AWT.Frame`

31. Consider the following fragment of code where strings `s1` and `s2` have been declared and assigned values:

   ```java
   JPanel jp = new JPanel();
   jp.setLayout(new BorderLayout());
   jp.add(new JButton(s1), BorderLayout.NORTH);
   jp.add(new JTextArea(12,12), BorderLayout.CENTER);
   jp.add(new JButton(s2), BorderLayout.SOUTH);
   ```

   When the panel `jp` is displayed on the screen, which of the following determines the width of the button at the top of the panel?

   A. The width of the string `s1`.
   B. The width of the `JTextArea`.
   C. The maximum of the widths of `s1` and the `JTextArea`.
   D. The maximum of the widths of `s2` and the `JTextArea`.
   E. The maximum of the widths of `s1`, `s2` and the `JTextArea`.

32. Consider the following fragment of code:

   ```java
   JFrame f = new JFrame("Test Frame");
   f.add(new JButton("Quit"), BorderLayout.CENTER);
   f.pack();
   f.setVisible(true);
   ```

   What is wrong with this code?

   A. The `JFrame` constructor does not take a `String` parameter.
   B. The default layout of a `JFrame` is not `BorderLayout`.
   C. The `getContentPane()` method should be used to add the button.
   D. `JFrame` does not have a `pack()` method.
   E. The `show()` method should be used instead of `setVisible()`.
33. Consider the following statements about modal dialogs in Java.
   1. A modal always has a Frame or another Dialog as owner.
   2. A modal dialog blocks user interaction with its owner when it is displayed.
   3. When a thread displays a modal dialog, execution of that thread is blocked until the dialog is closed.

Which of these statements are true?

A. 2 only.
B. 2 and 3 only.
C. 1 and 3 only.
D. 1 and 2 only.
E. All of them.

34. Consider the following statements about applets.
   1. An applet can open network connections to the HTTP server from which it was downloaded.
   2. An applet can read files from the local filestore.
   3. An applet can redirect the browser in which it is embedded to any URL.
   4. An applet can execute processes in the native operating system.

Which of these statements are true for an applet downloaded from a remote HTTP server?

A. 1 only.
B. 2 only.
C. 1 and 3 only.
D. 2 and 4 only.
E. 3 and 4 only.

35. Which of the following describes the effect of the flush() method in the java.io.BufferedOutputStream class?

A. The method is abstract and has no effect.
B. The method closes the output stream.
C. The method blocks any attempt to read from the stream until the buffer is full.
D. The method forces any bytes in the buffer to be written.
E. The method empties the buffer, losing any bytes that were in it.
36. What is the main difference between the BufferedReader and BufferedInputStream classes?

A. BufferedReader implements the FileReader interface, and BufferedInputStream does not.  
B. The BufferedReader class is a character stream, while the BufferedInputStream class is a byte-oriented stream.  
C. The BufferedReader class is an output stream, while the BufferedInputStream is an input stream.  
D. BufferedReader is an abstract class, which is implemented by the BufferedInputStream class.  
E. The BufferedInputStream class can be used for any input stream, but the BufferedReader class can only be used for file input.

37. Which of the following best describes the java.util.Properties class?

A. The class allows processes to be run in the native operating system.  
B. The class implements a tree structure that allows values to be retrieved efficiently.  
C. The class implements a persistent hash table storing key-and-value pairs as strings.  
D. The class provides access to the runtime context of an applet.  
E. The class provides access to parameters in the HTML file containing the URL of an applet.

38. Consider the following statements about the classes Socket and ServerSocket in the package java.net.

1. ServerSocket creates Socket instances.  
2. A Socket instance provides byte-oriented input and output streams.  
3. A ServerSocket will accept connections only from remote Java Virtual machines.  
4. ServerSocket.accept() blocks a thread until a connection is made.

Which of these statements are true?

A. 1, 3 and 4 only.  
B. 1, 2 and 4 only.  
C. 2, 3 and 4 only.  
D. 1 and 3 only.  
E. 2 and 4 only.
39. Which of the following best describes the role of stub objects in Remote Method Invocation?

A. A stub object is a local object that implements the same interface as the remote object; when one of its methods is called, it sends the parameters to the remote object, and receives the result of the corresponding method call in the remote object.  

B. A stub object is a local object that implements the same interface as the remote object; when one of its methods is called, it obtains the source code of the remote object, and compiles and runs the corresponding method.  

C. A stub object is a local copy of the state of the remote object that is sent across the network when one of the remote object’s methods is called.  

D. A stub object is the socket that is created to the remote machine on which the remote object is located.  

E. A stub object is the encoded result of the remote method invocation that is sent across the network.  

40. Which of the following statements about serialization are false?

A. Serialization is the process of translating object states into byte-streams.  

B. Serialization allows persistency: object states can be stored or sent across a network.  

C. Only classes that implement the Serializable interface can be used as arguments in Remote Method Invocation.  

D. The java.io.Serializable interface is empty.  

E. java.lang.Object implementsSerializable.