Advanced Object-Oriented Programming

TIME ALLOWED: Two hours

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
1. Which of the following best describes the meaning of ‘encapsulation’?

A. ‘Encapsulation’ means that every field in a class has public access and modifier methods. 
B. ‘Encapsulation’ means that a class is part of a package. 
C. ‘Encapsulation’ means that all fields in a class are private. 
D. ‘Encapsulation’ means that a class implements an abstract data type and hides implementation details. 
E. ‘Encapsulation’ means that the local state of an instance has been saved in encoded form.

2. Consider the following Java declarations.

```java
class AInt
{
    int val = 0;

    AInt() { val += 1; }
    AInt(int i) { val += i; }
}

class BInt extends AInt
{
    BInt() { val += 2; }
    BInt(int i)
    {
        val += i;
    }
}

class CInt extends BInt
{
    CInt(int i) { val += i; }
}
```

When an instance of CInt is created with `new CInt(3)`, what is the value of its field `val`?

A. 0 
B. 3 
C. 6 
D. 7 
E. 9
3. Suppose the following two classes are declared in separate files in the same directory:

```java
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. 3, 4 and 5 only
4. The following fragment of Java code will cause a compile-time error.

```java
JTextArea ta = new JTextArea(20,20);
JButton clear = new JButton("Clear");
clear.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        ta.setText(""");
    }
});
```

What is the cause of the compile-time error?

A. ActionListener is an interface and has no constructor.  
B. ActionListener should have only final methods.  
C. ta should be declared static.  
D. clear should be declared static.  
E. ta should be declared final.

5. Consider the following statements concerning a class Inner that is declared within a class Outer

1. Inner must be declared static.  
2. Inner must be declared static if it declares a static constant.  
3. If Inner is declared static, it can refer only to static members of Outer.  
4. If Inner is declared static, it must declare only static members.  
5. If Outer is declared static, then Inner must be declared static.

Which of these statements are true?

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

6. Which of the following best explains why interfaces allow for generic, polymorphic code?

A. Interfaces contain only abstract methods, which can be implemented in many different ways.  
B. All instances of all classes that implement an interface can be treated in the same way.  
C. Interfaces are used by the Thread class, so they allow many programs to run at the same time.  
D. Interfaces allow users to interact with programs by means of generic components such as buttons and menus.  
E. All instances of an interface share the same methods and fields.
7. Which of the following statements about interfaces is false?

A. An interface is an abstract class.
B. An interface may declare final variables.
C. An abstract class can inherit abstract methods from an interface by using the \texttt{implements} keyword.
D. An interface can inherit abstract method declarations from another interface by using the \texttt{extends} keyword.
E. An interface can declare static methods.

8. Consider the following fragment of Java code.

```java
class Outer{
    private class Inner1{
        int a;
        protected int b;
        private int c;
    }
    private class Inner2{ ...}
}
```

Which of the names \texttt{Inner1}, \texttt{a}, \texttt{b} and \texttt{c} are in scope within class \texttt{Inner2}?

A. All of them.
B. \texttt{Inner1}, \texttt{a} and \texttt{b} only.
C. \texttt{Inner1} and \texttt{a} only.
D. \texttt{Inner1} only.
E. None of them.

9. Which of the following is a true statement about javadoc comments?

A. Each line of a javadoc comment must begin with an asterisk.
B. All parameters to methods must be described within the \texttt{@param} tag.
C. The first sentence of a javadoc comment should be a brief summary of the class or member being described.
D. The first sentence of a javadoc comment should use the \texttt{@version} tag to specify the version of the JDK that is being used.
E. The first line of a javadoc comment should contain only the characters ‘/\*\*’. 
10. 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};
    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. Three.  
D. Four.  
E. Five.

11. 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, 2 and 4 only.  
E. 1, 2 and 3 only.
12. Which of the following statements about concurrency is false?
   A. Interference may arise when two or more threads access a shared resource.  
   B. Monitors ensure mutual exclusion of critical sections.  
   C. The Thread.wait() method allows a thread to surrender a monitor key.  
   D. An instance of a class with synchronized methods has one monitor key for each synchronized method.  
   E. Synchronized methods may lead to deadlock.

13. A thread can be in any of the following states: Ready, Running, Waiting, Blocked, and Dead. Which of the following statements is false?
   A. The notify() method may change a thread’s state from Waiting to Ready.  
   B. A thread becomes Dead when its run() method terminates.  
   C. The Thread.start() method puts a thread into the Running state.  
   D. The Thread.sleep() method changes a thread’s state from Running to Blocked.  
   E. Time-slicing can change a thread’s state from Running to Ready.

14. Why does Java no longer provide the Thread.stop() method?
   A. Calling the method usually resulted in deadlock.  
   B. The method was incompatible with the synchronized keyword.  
   C. The method compromised security in Applets.  
   D. To encourage programmers to write explicit routines to shut down threads.  
   E. To encourage programmers to use the synchronized keyword to ensure that all threads ended at the same time.

15. How is execution of a thread begun?
   A. By calling the Thread() constructor.  
   B. By creating a subclass of Thread() and calling the constructor for that class.  
   C. By calling the Thread.start() method.  
   D. By calling the Thread.init() method.  
   E. By calling the Thread.run() method.
16. Which of the following best describes ‘deadlock’?
   A. Deadlock arises when two or more threads concurrently access a shared resource.  
   B. Deadlock arises when two or more threads are given higher priority, and so prevent other threads from being executed.  
   C. Deadlock arises when two or more threads each control some resource, and so prevent the other threads from making progress.  
   D. Deadlock arises when a class contains a synchronized method, and so a lock is put upon the code in that method.  
   E. Deadlock arises when a thread’s `sleep()` method is invoked while that thread is in possession of a monitor key.

17. Which of the following is not characteristic of direct manipulation?
   A. Objects in the application domain are represented visually.  
   B. Results of actions are immediately visible.  
   C. Actions are rapid and reversible.  
   D. Interaction takes place through pointing and selecting.  
   E. Actions are controlled by a command language.

18. 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.

19. In the Model-View-Controller architecture, what does ‘model’ refer to?
   A. The user’s semantic model of the system  
   B. The programmer’s conceptual model of the system  
   C. The data that is manipulated by the system  
   D. The data that is output by the system  
   E. The best possible graphical representation of the system
20. 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. ☐

21. 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. ☐

22. An instance of which event class is created by a user entering a return character in a `TextField` component?

A. `TextFieldEvent` ☐

B. `TextComponentEvent` ☐

C. `TextChangedEvent` ☐

D. `ActionEvent` ☐

E. `ReturnEvent` ☐

23. What is the immediate superclass of `javax.swing.JComponent`?

A. `java.lang.Object` ☐

B. `java.AWT.Component` ☐

C. `java.AWT.Container` ☐

D. `java.AWT.Window` ☐

E. `javax.swing.Object` ☐
24. 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`  

25. 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`.  

26. Consider the following statements about modal dialogs in Java.

   1. A modal dialog 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.
27. What is the best way of setting up the graphical user-interface for an applet?
   
   A. Create and place all GUI components in the Applet constructor. □
   B. Create and place all GUI components using the appropriate tags in the HTML document. □
   C. Override the Panel constructor and invoke super(). □
   D. Override the Applet.init() method. □
   E. Override the Applet.start() method. □

28. 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. □

29. Consider the following Java command:

   ```java
   BufferedReader br =
   new BufferedReader(
       new InputStreamReader(System.in)
   );
   ```

Which of the following statements is false?

   A. The command allows lines of text to be read from standard input. □
   B. The command allows keyboard input to be read efficiently by storing characters in a buffer. □
   C. The command allows keyboard input to be written to a file. □
   D. The command converts the byte-oriented standard input stream to a character stream. □
   E. The command is an example of the use of wrapper classes. □
30. Which of the following statements best describes the difference between a byte stream and a character stream.

A. A byte stream contains only bytes, while a character stream contains only characters.  
B. A byte stream allows only single bytes to be written and read, while a character stream stores bytes in a buffer until there are enough to make up a character.  
C. A byte stream is an input stream, while a character stream is an output stream.  
D. A byte stream is used to read and write to memory, while a character stream is used to read keyboard input.  
E. A byte stream allows only single bytes to be written and read, while a character stream allows characters, integers and strings to be written and read.

31. Consider the following fragment of Java code:

```java
try {
    FileInputStream fis = ...;
    FileOutputStream fos = ...;
    ...
} catch (IOException ioe){}
finally {
    fis.close();
    fos.close();
}
```

Which of the following statements describes the effect of the `finally`-block?

A. The code will close the file I/O streams, whether or not an exception is thrown.  
B. The code will close the file I/O streams only if an exception is thrown.  
C. The code will close the file I/O streams only if an exception is thrown that is not of type `IOException`.  
D. The code will close the file I/O streams only if no exceptions are thrown.  
E. The code will not compile because `finally` is not a keyword in Java.

32. Which of the following fragments of code will allow strings to be added to a file called ‘testoutput’?

A. `File f = new File("testoutput");
f.println();`
B. `File f = new File("testoutput");
FileInputStream fis = new FileInputStream(f);`
C. `File f = new File("testoutput");
FileWriter fw = new FileWriter(f);`
D. `File f = new File("testoutput");
OutputStream os = f.getOutputStream();`
E. `File f = new File("testoutput");
InputStream is = f.getInputStream();
FileOutputStream fos = new FileOutputStream(is);`
33. Which of the following best describes a ‘socket’?

A. A socket is a connection across a network between a server process and a client process; only the server process can transmit data.

B. A socket is a connection across a network between two processes, and allows both processes to transmit data to each other.

C. A socket is a connection between two processes running on the same computer, and allows both processes to transmit data to each other.

D. A socket is a connection between two processes running on the same computer, and pipes data from one process to the other.

E. A socket is an abstract address for a port that is running a network process.

34. 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.

35. In Remote method Invocation, what is meant by ‘marshalling’?

A. Marshalling is the process of synchronizing all parameters to the method being invoked, in order to prevent data corruption arising from the concurrent method call.

B. Marshalling is the process of sending a copy of the remote object to the client JVM in order to invoke its method locally.

C. Marshalling is the process of sending the state of the client JVM on a byte stream to the remote object in order that the JVM can be started as a separate thread on the remote host.

D. Marshalling is the process of suspending the current thread in the client JVM until a result is obtained from the remote method invocation.

E. Marshalling is the process of sending the state of all parameter instances on a byte stream to the remote object.
36. Which of the following statements about stub objects in Remote Method Invocation is false?

A. A stub object is received by calling the `lookup()` method in the `java.rmi.Naming` class.
B. A stub object implements the same interface as the remote object.
C. A stub object serializes parameters to the remote method call.
D. A stub object implements the `RMISecurityManager` interface.
E. A stub object deserializes the result of a remote method call.

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 fragment of Java code.

```java
interface Comparable<A> {
    public A greater(A x, A y);
}

class Comparison {
    public <A, B extends A, C extends Comparable<A>>
    A test(A x, B y, C z)
    {
        return z.greater(x,y);
    }
}
```

Which of the following statements is true?

A. The code will cause a compile-time error because `A` cannot be used as a return type.
B. The code will cause a compile-time error because the parameter type `B` cannot extend the parameter type `A`.
C. The code will cause a compile-time error because `greater` requires its arguments to be of the same type.
D. The third argument to the method `test()` must be an instance of some class that implements the `Comparable<Aaa>` interface, where `Aaa` is the type of the first argument of `test()`.
E. The third argument to the method `test()` must be an instance of a class that explicitly extends a class that implements the `Comparable` interface by adding a method that takes two arguments of different types.
39. Java implements generic types by *erasure*: what does this mean?

A. The compiler removes references to parameter types and inserts casts wherever return types refer to parameter types.
B. The programmer must comment out references to parameter types before code is compiled.
C. The compiler removes any casts that may be type-unsafe.
D. The compiler removes any references to parameter types and inserts references to the closest superclass that will make expressions type-safe.
E. Any type-unsafe casts to parameter types will be ignored by the interpreter at run-time.

40. Consider the following class declaration.

```java
class Adder<A, B extends List<A>>
{
    Adder(A elt, B list) { ... }
}
```

Given an instance `e` of class `Integer` and `l` of class `List<Integer>`, which of the following is a correct call of this class’s constructor?

A. `<Integer, List<Integer>> Adder(e, l)`
B. `<A, B extends List<A>> Adder(e, l)`
C. `<Integer> Adder(e, l)`
D. `Adder<Integer, List<Integer>>(e, l)`
E. `Adder<Integer, B extends List<Integer>>(e, l)`