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. For a given class and abstract data type, which of the following best describes the meaning of ‘adequacy of representation’?
   A. 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. ☐
   B. Every operation in the abstract data type is implemented by a method in the class. ☐
   C. Each public method in the class implements one of the operations in the abstract data type. ☐
   D. The class has no public constructor, but provides public methods that create instances of the class. ☐
   E. The class has a field that stores values of the abstract data type. ☐

3. 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. ☐

4. Which of the following is a true statement about javadoc comments?
   A. Each line of a javadoc comment must begin with an asterisk. ☐
   B. Each line of a javadoc comment should end with the @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 begin with the @param tag. ☐
   E. The first line of a javadoc comment should contain only the characters ‘/**’. ☐
5. Suppose a class Inner is declared within a class Outer; which of the following statements is false?
   A. All members of Outer are visible within Inner.
   B. Private methods of Inner are not visible in any other classes declared within Outer.
   C. Inner must be declared static.
   D. static members can be declared within Inner.
   E. Inner can be declared public, private, or protected.

6. Which of the following is a true statement about anonymous classes?
   A. An anonymous class can be declared public, protected or private.
   B. An anonymous class can be declared static.
   C. An anonymous class must be declared final.
   D. An anonymous class is always an implementation of an interface.
   E. An anonymous class is always an inner class.

7. 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. There should not be a semicolon after getValue().
   C. There should not be a semicolon on the second-last line.
   D. The variable i should be declared final.
   E. The variable i should be declared static.
8. Consider the following code:

```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. There is no body for the method getValue() in the interface Number.  
C. There should be a semicolon at the end of the declaration of Zero.  
D. The field zero is declared static, so the class Zero should be declared static.  
E. The field zero is declared final, so the class Zero should be declared final.

9. 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 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.  
C. 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.  
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.
10. Which of the following statements about RuntimeException is true?

A. A RuntimeException does not need to be caught because the Java interpreter catches it automatically.

B. A RuntimeException should never be caught because there is no way to continue after such an exception is thrown.

C. A RuntimeException need not be advertised because it generally arises through an unforeseeable programming error.

D. A RuntimeException need not be advertised because it will only be thrown if the Java interpreter is interrupted by the user.

E. A RuntimeException can never be caught; it should always be advertised in a throws-clause.

11. Which of the following is a true statement about time-slicing in Java?

A. Time-slicing can only be ensured if every thread calls the sleep() method.

B. Time-slicing can only be ensured if every thread calls the wait() and notify() methods.

C. Time-slicing requires a multiple-processor architecture.

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.

12. Which of the following best describes the effect of the sleep() method in a multithreaded program?

A. Calling sleep() causes execution of a thread to pause for a specified amount of time, after which execution of the thread immediately resumes.

B. Calling sleep() causes execution of a thread to pause for a specified amount of time, after which the thread is ready to run.

C. Calling sleep() causes execution of the thread to pause until another thread calls sleep().

D. Calling sleep() causes execution of the thread to pause until another thread calls the wake() method.

E. Calling sleep() causes execution of the thread to end.

13. In the context of multithreaded programs, what is the purpose of a ‘monitor’?

A. A monitor ensures that at most one thread is running a critical section at any time.

B. A monitor implements time-slicing.

C. A monitor is code that is invoked whenever deadlock occurs.

D. A monitor ensures that all threads are allowed to run for the same amount of time.

E. A monitor prevents threads calling methods in other threads.
14. In the context of multithreaded programs, what is ‘interference’?
   A. Interference is when one thread calls the `interrupt()` method in 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 the technique of throwing an exception in one thread and catching it in another. □
   E. Interference is data corruption arising from several threads accessing a shared resource concurrently. □

15. Consider the following fragment of code:

```
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 cause a compile-time error because `new Integer(0)` is not static. □
   B. The code will cause a compile-time error because `System.out` is not static. □
   C. 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. □
   D. 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. □
   E. The code will compile, but it will almost certainly cause deadlock. □

16. Thread-safety can be ensured in Java by qualifying a method with the keyword `synchronized`. This ensures that synchronized method 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. There is one key for each synchronized method in the class, and these keys are shared by all instances of the class. □
   D. There is one key, shared by all instances of the class, regardless of the number of instances of the class. □
   E. Each instance of each class that can call the synchronized method has its own key. □
17. Which of the following best describes the effect of calling the `wait()` method inside a synchronized method?

A. 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.

B. 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.

C. 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.

D. When `wait()` is invoked in a thread, that thread is given highest priority, and all other threads must wait for that thread to terminate.

E. When `wait()` is invoked in a thread, all other threads must wait until execution of the synchronized method terminates.

18. 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.

19. 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.

20. 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.
21. 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.  

22. Which of the following describes ‘Knowledgeable, Intermittent Users’?
   A. High levels of both syntactic and semantic knowledge.  
   B. A high level of syntactic knowledge, but limited semantic knowledge.  
   C. Limited syntactic knowledge, but a high level of 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.  

23. Which form of interaction is generally best suited to expert users?
   A. Form fill-in.  
   B. Menu selection.  
   C. Command language.  
   D. Direct manipulation.  
   E. Touch-screen.  

24. Which of the following statements about the java.AWT and javax.swing packages is true?
   A. All components in both packages are thread-safe.  
   B. Most components in both packages are thread-safe.  
   C. Most components in java.AWT are thread-safe, but most in javax.swing are not.  
   D. Most components in javax.swing are thread-safe, but most in java.AWT are not.  
   E. No components in either package are thread-safe.  

25. Which of the following best describes the Model-View-Controller architecture?
   A. Data, display, and user controls are implemented in separate classes.  
   B. Data, display, and user controls are implemented in separate packages.  
   C. Data is implemented in one class, and there is a clear separation between the implementation of functionalities for display and data manipulation.  
   D. Data is implemented in a separate package, 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. There is a special thread that handles all AWT events.  
B. Each AWT event causes a new thread to be created to handle that event.  
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. 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

28. 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.

29. 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

---

PAPER CODE COMP213  page 9 of 13  Continued
30. 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().

31. What does it mean to say that a pop-up Dialog is ‘modal’?

A. The Dialog can be re-used.  
B. The Dialog can perform more than one action.  
C. Other window components are inactive while the Dialog is visible.  
D. The Dialog still exists while it is not visible.  
E. The user can alter the size of the Dialog.

32. Which of the following is true for an applet downloaded from a remote HTTP server?

A. The applet can read and write to files on the native file system.  
B. The applet can only read files on the native file system, but cannot write to them.  
C. The applet can open network connections to any HTTP server.  
D. The applet can only open network connections to the HTTP server from which it was downloaded.  
E. The applet can execute processes in the native operating system.

33. 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 blocks any attempt to read from the stream until the buffer is full.  
C. The method forces any bytes in the buffer to be written.  
D. The method empties the buffer, losing any bytes that were in it.  
E. The method closes the output stream.
34. What is the main difference between the BufferedWriter and BufferedOutputStream class?

A. The BufferedWriter class is a character stream, while the BufferedOutputStream class is a byte-oriented stream. □
B. The BufferedWriter class is an input stream, while the BufferedOutputStream is an output stream. □
C. The BufferedWriter class has a larger buffer. □
D. BufferedWriter is an abstract class, which is implemented by the BufferedOutputStream class. □
E. The BufferedOutputStream class can be used for any output stream, but the BufferedWriter class can only be used for file output. □

35. 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. □

36. 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); □
37. 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.

38. 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.
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. Consider the following three statements about serialization.

1. Serialization is the process of translating object states into byte-streams.
2. Serialization allows persistency: object states can be stored or sent across a network.
3. Only classes that implement the `Serializable` interface can be used in Remote Method Invocation.

Which of these three statements are true?

A. 1 only.

B. 1 and 2 only.

C. 1 and 3 only.

D. 2 and 3 only.

E. 1, 2 and 3.