Question point values are shown in parenthesis.

1. (20) For each of the following statements, indicate if the statement is true or false.

   a. The use of abstract ***classes*** enhances your ability to take advantage of polymorphic behavior within a class hierarchy.

   b. The use of abstract ***methods*** enhances your ability to take advantage of polymorphic behavior within a class hierarchy.

   c. A class that inherits an abstract method does not have to provide detail for the inherited method. However, if it does not provide detail, then it is ***impossible*** to instantiate objects from that class.

   d. When an object is created, a constructor for the object's superclass is ***always*** called.

   e. While a class may only extend a single class, there is ***no limit*** on the number of interfaces it may implement.

2. (16) Java uses interfaces for two distinct reasons. What are these two reasons?

3. (16) How are Vectors and Hashtables similar? How are they different?

4. (16) You have the following class definition:

```
package preferences;

private class Preferences
{
    private String URLInfo;
    public Preferences(String url)
    {
        URLInfo = url;
    }
    public void setURL(String url)
    {
        URLInfo = url;
    }
    public String getURL()
    {
        return URLInfo;
    }
}
```

   Indicate the changes necessary to be able to serialize this class.

5. (16) Contrast the similarities/differences between a Java Applet and Java Servlet.

6. (16) Assume you have a user interface that contains a JTextField component named txtAge. Write a segment of Java code that stores the contents of the text field into the variable “age” whose type is an int. Your code should display an error message using a JOptionPane component if the content of the text field contains an invalid number.

IS 423 – Aut 00
Exam 2 Answer Key

1. All parts (a-e) are true.

2. An interface provides two functions within the Java language. First, it provides an equivalent of multiple inheritance. In this regard, it forces classes that implement it to provide details for the abstract methods defined in the interface. This means that the implementing class “is-a” interface.

   The second reason for interfaces is to ensure that the implementing class really wants the functionality defined by the interface. A good example of this is the Serializable interface. This interface has no abstract methods for the implementing class to define, but classes the implement the Serializable interface can be serialized by the java.io classes.

3. Vectors and Hashtables are similar in that they can store an unlimited number of references to any types of objects.

   They are different in the way they store/retrieve the objects. Vectors use an indexing system while Hashtables use a hashing scheme based on the value of a “key” field.

4. Changes are shown in bold print:

```
package preferences;

import java.io.*;
public class Preferences implements Serializable
{
    private String URLInfo;
    public Preferences() {}
    public Preferences(String url)
    {
        URLInfo = url;
    }
```

5. Applets and Servlets are similar in that they both are written using the Java language. They differ in where they execute – Applets execute within the client browser while Servlets execute on the server.

6.
```
int age = 0;
try
{
    age = Integer.parseInt(txtAge.getText());
}
catch (NumberFormatException e)
{
    JOptionPane.showMessageDialog(this, "Number not valid");
}
```