CS 201
Final Examination
December 15, 2004
Total points possible- 75

Student ID number:
Student Name:

10 am – 12 pm
 Obtained Score:

Instructions:
This exam is a closed book, no notes, no calculator exam. There are 10 questions total. For all questions write your answers on the exam and hand in all pages of the exam. There should be 8 pages including this page, please check you have all pages before you begin. If you use the back of the pages for your answers, make sure you indicate that on the front side.

Please write your student number and name at the top of this page, and sign in the space provided.

Note: You may not leave during the last 15 minutes of the exam.
Good luck!

Question 1 – 5 points
Question 2 – 5 points
Question 3 – 5 points
Question 4 – 5 points
Question 5 – 15 points
Question 6 – 15 points
Question 7 – 10 points
Question 8 – 15 points

Total: 75 points
1) True or False, when String objects are compared with ==, the result is true if the Strings contain the same values. 5 points

Solution:
False. String objects that are compared with operator == are actually compared to determine if they are the same object in memory. Use the equals() method to compare strings.

2) List common examples of exceptions. 5 points

Solution:
Memory exhaustion, array subscript out of bounds, arithmetic overflow, division by zero, invalid method parameters.

3) Given a superclass and a subclass, discuss the access rights of public, private, and protected members. 5 points

Solution:
A superclass’s public members are accessible anywhere the program has a reference to that superclass type or one of its subclass types. A superclass’s private members are accessible only in methods of that superclass. A superclass’s protected access members serve as an intermediate level of protection between public and private access. A superclass’s protected members may be accessed only by methods of the superclass, by methods of subclasses and by methods of other classes in the same package (protected members have package access).

4. Describe through an example the Object Oriented concept of Inheritance. Make sure to explain the relationships between your objects.

5 points

Solution:
For example, mountain bikes, racing bikes, and tandems are all kinds of bicycles. Mountain bikes, racing bikes, and tandems are all subclasses of the bicycle class. Similarly, the bicycle class is the superclass of mountain bikes, racing bikes, and tandems.

5. Write a method countdown that takes a single integer as a parameter from the user. If the parameter is zero, it prints the word “Blastoff”. Otherwise, it prints the
number and then invokes a method names countdown – itself – passing an n-1 as an argument.

15 points

Solution:

```java
import java.io.*;

public class countdown {
    public static void countdown(int n) {
        if (n == 0) {
            System.out.println("Blastoff!");
        } else {
            System.out.println(n);
            countdown(n - 1);
        }
    }

    public static void main(String args[]) throws IOException {
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Please enter a number: ");
        int input = Integer.parseInt(stdin.readLine());
        countdown(input);
    }
}
```

6. Write a recursive method to determine the sum of the digits of an integer. For example, the sum of digits of 51624 is 5 + 1 + 6 + 2 + 4 = 18

15 points

Solution:
Import java.io.*;
Public class SumofDigits
{
    Public static int add (int sum)
    {
        If (num<10)
        {
            Return num;
        }
        Else
        {
            Return ((num%10) + add(num/10));
        }
    }
    Public static void main (String args []) throws IOException
    {
        BufferedReader stdin = new BufferedReader (new InputStreamReader (System.in));

        System.out.println ("Please enter a number:" );
        Int input = Integer.parseInt (stdin.readline () );
        System.out.println ("The sum of the digits of " + input + " is " + add (input));
    }
}

7. Name and describe 4 of the functions that commonly must be overridden when working with Applets.

10 points

Solution:
init( ) – Initialization when applet is loaded.
start( ) – Where the applet start after it is initialized.
stop( ) – Runs when the user exits the page.
paint( ) – What applet uses to display something on screen.
destroy( ) – Allows the applet to clean up after itself.

8) Write a complete Java applet to prompt the user for the floating-point radius of a sphere, and call method sphereVolume to calculate and display the volume of that sphere using the assignment:

    volume = (4.0/3.0) * Math.PI * Math.pow(radius, 3)

The user should input the radius through a JTextField.
15 points

Solution.
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SphereTest extends JApplet implements ActionListener
{
    JLabel promptLabel;
    JTextField inputField;

    public void init()
    {
        Container container = getContentPane();
        container.setLayout(new FlowLayout());

        promptLabel = new JLabel("Enter sphere radius: ");
        inputField = new JTextField(10);
        inputField.addActionListener(this);
        container.add(promptLabel);
        container.add(inputField);
    }

    public void actionPerformed(ActionEvent actionEvent)
    {
        double radius =
            Double.parseDouble(actionEvent.getActionCommand());

        showStatus("Volume is " + sphereVolume(radius));
    }

    public double sphereVolume(double radius)
    {
        double volume =
            (4.0/3.0) * Math.PI * Math.pow(radius,3);

        return volume;
    }
}