

Name:

1.124 Quiz 2

Time: 1 hour 20 minutes

Answer all questions. All questions carry equal marks.

Thursday November 9, 2000

Question 1.

Show the steps that are involved in sorting the string *SORTME* using the quicksort algorithm given below.

```
#include <iostream.h>
void quicksort(char *a, int l, int r);

main() {
    char str[8] = "9SORTME";    // 9 is a sentinel.
    quicksort(str, 1, 6);
}

inline void swap(char *a, int i, int j) {
    char tmp = a[i];
    a[i] = a[j];
    a[j] = tmp;
    cout << a+1 << endl;    // Print out the array, excluding the sentinel.
}

void quicksort(char *a, int l, int r) {
    if (r > l) {
        char v = a[r];
        int i = l - 1;
        int j = r;
        while (1) {
            while(a[++i] < v);
            while(a[--j] > v);
            if (j <= i)
                break;
            swap(a, i, j);
        }
        swap(a, r, i);

        quicksort(a, l, i-1);
        quicksort(a, i+1, r);
    }
}
```

Answer:

S	O	R	T	M	E

Question 2.

Show how you would translate the bold portions of the following C++ code into Java.

```
#include <iostream.h>
```

```
class Shape {
```

```
private:
```

```
    float x, y;
```

```
public:
```

```
    Shape(float a, float b) {
```

```
        x = a;
```

```
        y = b;
```

```
    }
```

```
    virtual float compute_area() = 0;
```

```
    virtual void print() {
```

```
        cout << x << " " << y << endl;
```

```
    }
```

```
};
```

```
class Circle : public Shape {
```

```
private:
```

```
    float radius;
```

```
public:
```

```
    Circle(float a, float b, float r) : Shape(a, b) {
```

```
        radius = r;
```

```
    }
```

```
    float compute_area() {
```

```
        return 3.14f * radius * radius;
```

```
    }
```

```
    void print() {
```

```
        cout << radius << endl;
```

```
        Shape::print();
```

```
    }
```

```
};
```

```
void main() {
```

```
    Circle a(3,4,2);
```

```
    a.print();
```

```
}
```

Answer:

Question 3.

In the following C++ program, the *outputData()* function can handle callbacks such as *plot()* and *print()*. How would you complete the given Java code to implement a similar capability?

```
#include <iostream.h>
```

```
class Point {
```

```
private:
```

```
    int x, y;
```

```
public:
```

```
    Point(int a = 0, int b = 0) {
```

```
        x = a;
```

```
        y = b;
```

```
    }
```

```
    void print() {
```

```
        cout << x << " " << y << endl;
```

```
    }
```

```
};
```

```
typedef void (*OutFunc)(Point& p);
```

```
void plot(Point &p) {
```

```
// Assume that this plots the point p on the screen.
```

```
    cout << "In plot:" << endl;
```

```
    p.print();
```

```
}
```

```
void print(Point &p) {
```

```
// Assume that this prints out the coordinates of p.
```

```
    cout << "In print:" << endl;
```

```
    p.print();
```

```
}
```

```
void outputData(OutFunc pFunc, Point *a, int n) {
```

```
    for (int i = 0; i < n; i++)
```

```
        pFunc(a[i]);
```

```
}
```

```
void main() {
```

```
    Point a[2];
```

```
    a[0] = Point(2,3);
```

```
    a[1] = Point(4,5);
```

```
    outputData(plot, a, 2);
```

```
    outputData(print, a, 2);
```

```
}
```

Answer:

```
class Point {
    private int x, y;

    public Point(int a, int b) {
        x = a;
        y = b;
    }

    void print() {
        System.out.println(x + " " + y);
    }
}

class Plotter
{

}

class Printer
{

}

class Main {
    static void outputData(
    ) {

    }

    public static void main(String args[]) {
        Point a[] = new Point[2];
        a[0] = new Point(2,3);
        a[1] = new Point(4,5);

        outputData(new Plotter(), a, 2);
        outputData(new Printer(), a, 2);
    }
}
```

Question 4.

Show how you would complete the given Java code, so that it achieves the effect shown in the Figure below.



Answer:

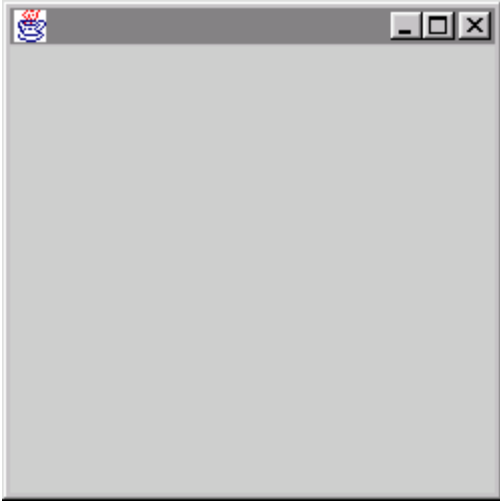
```
import java.awt.*;
import javax.swing.*;

class Main {
    public static void main(String args[]) {
        JFrame f = new JFrame();

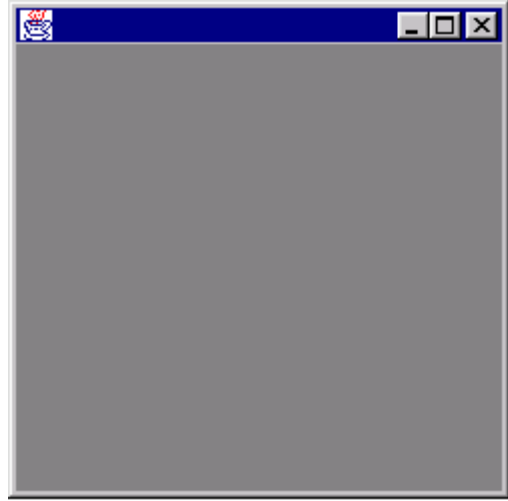
        f.setSize(250,250);
        f.setVisible(true);
    }
}
```

Question 5.

How you would you change the background color of the panel when the mouse moves over the application's window?



Mouse out



Mouse over

```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
  
class Main {  
    public static void main(String args[]) {  
        JFrame f = new JFrame();  
        final JPanel p = new JPanel();
```

Answer:

```
        f.setContentPane(p);  
        f.setSize(250,250);  
        f.setVisible(true);  
    }  
}
```

Question 6.

The following applet contains several errors. Explain what changes you would make to correct the code, so that the applet displays the current frame number.

Answer:

```
/*
  <APPLET CODE=MyApplet.class WIDTH=250 HEIGHT=100>
</APPLET>
*/

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

public class MyApplet extends JApplet
{
    Thread t = null;
    int count = 0;

    public void init() {
        getContentPane().add(new JPanel() {
            public void paintComponent(Graphics g) {
                super.paintComponent(g);
                g.drawString("Count = " + count, 50, 50);
            }
        });
    }

    public void start() {
        t = new Thread();
        t.start();
    }

    public void run() {
        for (int i = 0; i < 200; i++) {
            count++;
            repaint();
            Thread.sleep(100);
        }
    }
}
```

Question 7.

What is double buffering and why is it important in animation? How do the Swing classes differ from the AWT classes in this respect?

Answer:

Question 8.

Show how you would extract the number *1.124* from the string *"Hello 1.124 World!"* and then store it in a *float* variable.

Answer: