

CS106A Review Session Solutions

Written by Jeremy Keeshin

1. 1D Arrays

```
private int cycleLength(int[] arr, int startIndex){
    int cycleLength = 0;
    int curIndex = startIndex;

    while(true){
        int nextIndex = arr[curIndex];
        cycleLength++;
        if(nextIndex == startIndex) break;
        curIndex = arr[curIndex];
    }
    return cycleLength;
}
```

2. String Processing

```
private HashMap<String, String> getQueryParameters(String url){
    HashMap<String, String> params = new HashMap<String, String>();
    int queryStart = url.indexOf("?");
    String query = url.substring(queryStart+1);
    StringTokenizer st = new StringTokenizer(query, "&");
    while(st.hasMoreTokens()){
        String key = st.nextToken();
        String val = st.nextToken();
        val = val.replace('+', ' ');
        params.put(key, val);
    }
    return params;
}

private void PrintQueryParameters(HashMap<String, String> map){
    Set<String> keys = map.keySet();
    Iterator<String> it = keys.iterator();
    while(it.hasNext()){
        String key = it.next();
        String val = map.get(key);
        println(key + " : " + val);
    }
}
```

3. 2D Arrays

```
private int[] grade(char[][] grid){
    char[] solutions = grid[0];
    int[] results = new int[grid.length-1];
    for(int i = 1; i < grid.length; i++){
        results[i-1] = gradeRow(grid[i], solutions);
    }
    return results;
}

private int gradeRow(char[] answers, char[] solutions){
    int score = 0;
    for(int i = 0; i < answers.length; i++){
        if(answers[i] == solutions[i])
            score++;
    }
    return score;
}
}
```

4. Graphics Programs and Mouse Events

```
public class ClickCircle extends GraphicsProgram{

    private static final double DELTA_MOVE = 10;
    private static final double DIAMETER = 80;
    private static final double RADIUS = DIAMETER / 2.0;
    private static final int PAUSE_TIME = 80;
    private static final int NUM_STEPS = 15;
    private static final double ERROR = 5;

    //ivars
    private GOval circle;
    private double dx, dy;
    private GPoint nextPoint;
    private RandomGenerator rgen = RandomGenerator.getInstance();

    public void init(){
        circle = new GOval(DIAMETER, DIAMETER);
        circle.setFilled(true);
        circle.setColor(Color.black);
        double startX = rgen.nextDouble(0, getWidth() - DIAMETER);
        double startY = rgen.nextDouble(0, getHeight() - DIAMETER);
        add(circle, startX, startY);
        dx = 0;
        dy = 0;
        nextPoint = null;
        addMouseListeners();
    }

    public void mouseClicked(MouseEvent e){
        double nextX = e.getX();
```

```

        double nextY = e.getY();
        dx = (nextX - circle.getX()) / NUM_STEPS;
        dy = (nextY - circle.getY()) / NUM_STEPS;
        nextPoint = new GPoint(e.getPoint());
    }

    public void run(){
        while(true){
            circle.move(dx, dy);
            pause(PAUSE_TIME);
            if(reachedEndpoint()) break;
        }

        GLabel gameOver = new GLabel("Game Over");
        gameOver.setFont("Georgia-50");
        add(gameOver, getWidth()/2 - gameOver.getWidth()/2,
            getHeight()/2 - gameOver.getAscent()/2);
    }

    private boolean reachedEndpoint(){
        if(nextPoint == null) return false;
        double xDist = Math.abs(circle.getX() + RADIUS -
            nextPoint.getX());
        double yDist = Math.abs(circle.getY() + RADIUS -
            nextPoint.getY());
        double dist = Math.sqrt(xDist*xDist + yDist*yDist);
        return dist <= ERROR;
    }
}

```

5. Data Structures and Large Program Design

Note: This program is larger than what you will see on the final, but many parts of it are things you could write.

```

public class MicrosoftOffice1995 extends ConsoleProgram{

    private JTextField userField, toField, fromField,
    private JTextField subjectField, messageNumberField;
    private JTextArea messageArea;
    private JButton send, clear, reply;
    private static final int TEXT_LENGTH = 20;
    private static final int TEXT_HEIGHT = 30;
    private String SPACER = "===== ";
    private static final String FILE = "emails.txt";
    private String MESSAGE_END = "####";

    private HashMap<String, ArrayList<MyMessage>> emailInfo;
    private String currentUser;

    public void init(){
        setSize(800, 800);
        addMessageBar();
        addTopBar();
        addActionListeners();
        setFont("Courier-15");
    }
}

```

```

        emailInfo = new HashMap<String, ArrayList<MyMessage>>();
    }

    public void run(){
        readFile();
    }

    private MyMessage readMessage(BufferedReader rd){
        try{
            String from = rd.readLine();
            if(from == null) return null;
            String to = rd.readLine();
            String subj = rd.readLine();
            String message = "";
            while(true){
                String cur = rd.readLine();
                if(cur == null || cur.equals(MESSAGE_END)) break;
                message += cur + "\n";
            }
            MyMessage m = new MyMessage(to, from, subj, message);
            return m;
        }catch(IOException ex){
            return null;
        }
    }

    private void readFile(){
        try {
            BufferedReader rd = new BufferedReader(new
                FileReader(FILE));
            while(true){
                MyMessage m = readMessage(rd);
                if(m == null) break;
                String to = m.getRecipient();
                ArrayList<MyMessage> inbox;
                if(!emailInfo.containsKey(to)){
                    inbox = new ArrayList<MyMessage>();
                    emailInfo.put(to, inbox);
                }
                inbox = emailInfo.get(to);
                inbox.add(m);
            }
            rd.close();
        } catch (IOException e) {

        }
    }

    private void printMessagePreview(MyMessage m){
        String from = m.getSender();
        String subj = m.getSubject();
        println("FROM: " + from);
        println("SUBJ:" + subj);
    }

```

```

private void printInbox(){
    ArrayList<MyMessage> inbox = emailInfo.get(currentUser);
    fromField.setText(currentUser);

    if(inbox == null){
        inbox = new ArrayList<MyMessage>();
        emailInfo.put(currentUser, inbox);
    }

    if(inbox.size() == 0) println("Empty inbox!");
    else{
        for(int i = 0; i < inbox.size(); i++){
            println(SPACER);
            println("Message " + i);
            printMessagePreview(inbox.get(i));
        }
    }
}

private void switchUser(){
    currentUser = userField.getText();
    println(SPACER);
    println("The current inbox is for user " + currentUser);
    printInbox();
}

private MyMessage createEmail(){
    String to = toField.getText();
    String subj = subjectField.getText();
    String message = messageArea.getText();
    MyMessage m = new MyMessage(to, currentUser, subj, message);
    return m;
}

private void sendEmail(){
    MyMessage m = createEmail();
    if(!emailInfo.containsKey(m.getRecipient())){
        ArrayList<MyMessage> inbox = new ArrayList<MyMessage>();
        emailInfo.put(m.getRecipient(), inbox);
    }
    ArrayList<MyMessage> oldInbox =
        emailInfo.get(m.getRecipient());
    oldInbox.add(m);
    emailInfo.put(m.getRecipient(), oldInbox);
    clearMessageBar();
}

private void clearMessageBar(){
    toField.setText("");
    fromField.setText("");
    subjectField.setText("");
    messageArea.setText("");
}

private void displayMessage(){

```

```

        int messageNumber =
            Integer.parseInt(messageNumberField.getText());
        ArrayList<MyMessage> inbox = emailInfo.get(currentUser);
        MyMessage m = inbox.get(messageNumber);

        toField.setText(m.getRecipient());
        fromField.setText(m.getSender());
        subjectField.setText(m.getSubject());
        messageArea.setText(m.getMessage());
    }

    private void reply(){
        String to = fromField.getText();
        String subj = subjectField.getText();
        fromField.setText(currentUser);
        toField.setText(to);
        subjectField.setText("RE: " + subj);
        messageArea.setText("\n\n\n>>" + messageArea.getText());
    }

    public void actionPerformed(ActionEvent e){
        if(e.getSource() == send){
            sendEmail();
        }
        if(e.getSource() == userField){
            switchUser();
        }
        if(e.getSource() == messageNumberField){
            displayMessage();
        }
        if(e.getSource() == clear){
            clearMessageBar();
        }
        if(e.getSource() == reply){
            reply();
        }
    }

    private void addMessageBar(){
        toField = new JTextField(TEXT_LENGTH);
        fromField = new JTextField(TEXT_LENGTH);
        subjectField = new JTextField(TEXT_LENGTH);
        messageArea = new JTextArea(TEXT_LENGTH, TEXT_HEIGHT);
        send = new JButton("SEND");
        clear = new JButton("CLEAR");
        reply = new JButton("REPLY");
        add(clear, WEST);
        add(reply, WEST);
        add(new JLabel("FROM: "), WEST);
        add(fromField, WEST);
        add(new JLabel("TO: "), WEST);
        add(toField, WEST);
        add(new JLabel("SUBJECT: "), WEST);
        add(subjectField, WEST);
        add(messageArea, WEST);
        add(send, WEST);
    }

```

```

    }

    private void addTopBar(){
        JLabel view = new JLabel("View Inbox For: ");
        userField = new JTextField(TEXT_LENGTH);
        add(view, NORTH);
        add(userField, NORTH);
        add(new JLabel("Message number: "), NORTH);
        messageNumberField = new JTextField(5);
        add(messageNumberField, NORTH);
        messageNumberField.addActionListener(this);
        userField.addActionListener(this);
    }
}

public class MyMessage {

    String sender, recipient, subject, message;

    public MyMessage(String to, String from, String sub,
        String mes){
        sender = from;
        recipient = to;
        subject = sub;
        message = mes;
    }

    public String getRecipient(){
        return recipient;
    }

    public String getSender(){
        return sender;
    }

    public String getSubject(){
        return subject;
    }

    public String getMessage(){
        return message;
    }
}

```