



ICS3U **Sample** Final Exam
Introduction to Computer Science

Tuesday January 28, 2025
Period 4 Exam Day, Room 129
Ms. Gorski

7 Question Pages
20 Questions
School Time Allotted: 8:30 – 11:00
Expected Time to Write Exam: 1.5 hours

Name (first and last): _____

Signature: _____

Total: / **130** %

Instructions:

- Students should answer all questions directly on the exam paper.
- Students may write in pencil or black pen or dark blue pen.
- Extra paper is permitted, but probably won't be needed.
- Water is permitted.
- Additional materials, such as bags, coats, and phones, are not permitted at the student's desk during the exam. They may be left at home, or in the student's locker, or at the front of the room.
- No phones or other electronic devices are permitted. Smart watches, headphones and earbuds are not permitted at the student's desk or in their pockets. These devices must be left in the student's bag at the front of the room.

1. Translate 18 from our number system to binary.

| | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 16 | 8 | 4 | 2 | 1 |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

/2

2. The following questions are based on this class.

```
public class ifs
{
    public static void main (String args[])
    {
        new ifs ();
    }

    public ifs ()
    {
        int num = IO.inputInt ("Number? ");
        if (num == 1)
            System.out.print ("Apple");
        else if (num > 30)
            System.out.print ("Banana");
        else if (num == 14)
            System.out.print ("Lime");

        if (num < 3)
            System.out.print ("Orange");
        else
            System.out.print ("Peach");
    }
}
```

Identify the following in the code:

| | | |
|-----|------------------------------------|----------------------|
| (a) | What is the class name? | <input type="text"/> |
| (b) | Identify a Boolean expression. | <input type="text"/> |
| (c) | What type of variable is declared? | <input type="text"/> |
| (d) | What is the variable name? | <input type="text"/> |
| (e) | What should the file be saved as? | <input type="text"/> |

/10

What is printed for each of the following inputs?

| | |
|-----------|----------------------|
| num is 1 | <input type="text"/> |
| num is 30 | <input type="text"/> |
| num is 31 | <input type="text"/> |
| num is 14 | <input type="text"/> |
| num is -2 | <input type="text"/> |

3. (a) What are the parts of a loop?

/8

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | I | L | S | V | | |
| 2 | T | L | S | C | | |
| 3 | S | T | R | | | |
| 4 | P | T | T | L | S | C |

(b) On **both** of the following loops, use the numbers from (a) to label each part.

```




for(  int i=0 ;  i<100 ;  i++ ) 
    System.out.print("The leaves are falling.");
System.out.println();

```

```

char stop = 'n'; 
while(  stop=='n' ) { 
    System.out.println("Today is a sample test.");
    System.out.println("How exciting. How wonderful.");
    System.out.println("Who doesn't love a SAMPLE TEST? Hooray!");
    stop = IO.inputChar("Again? (y/n) "); 
}

```

4. These questions relate to the PDLC. /9

(a) What does PDLC stand for?

| |
|---|
| P |
| D |
| L |
| C |

(b) What are the phases of the PDLC (in order)?

| | |
|---|--|
| 1 | |
| 2 | |
| 3 | |
| 4 | |

(c) Name the associated phase of the PDLC.

| | | |
|---|---------------------------------|--|
| 1 | Candy Crush Key Innovation | |
| 2 | Dumb Ways To Die Key Innovation | |
| 3 | Beta Testing. | |
| 4 | Write if statements. | |
| 5 | Write comments. | |
| 6 | Draw a structure chart. | |

5. Circle the most correct answer concerning the method shown (true or false). /5

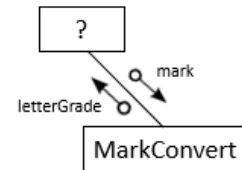
```
public String fizzBuzz (int n){
    if(n%15==0)
        return "FizzBuzz";
    else if (n%3==0)
        return "Fizz";
    else if (n%5==0)
        return "Buzz";
    else
        return n+"";
}
```

- T F a) The method name is fizzBuzz.
- T F b) The method return type is int.
- T F c) The parameter type is n.
- T F d) The parameter name is fizzBuzz.
- T F e) This method would send out a String.

6. Write a method that takes an integer and returns a char. /9

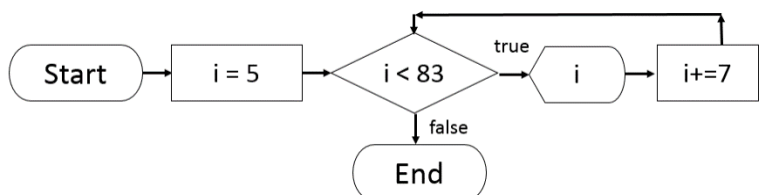
The char should be the A, B, C, D or F that corresponds with the mark. Nothing should be printed out in the method, return the char instead.

| | | | | | |
|--------|-----|-------|-------|-------|------|
| Letter | A | B | C | D | F |
| Mark | 80+ | 70-79 | 60-69 | 50-59 | 49-0 |



```
_____ ( _____ ) {
    method start word      return type (output)      method name (see structure chart)      parameter type      parameter name (input)
    if ( _____ )
        _____ ;
    else if ( _____ )
        _____ ;
    else if ( _____ )
        _____ ;
    else if ( _____ )
        _____ ;
    else
        _____ ;
}
```

7. Write for loop code that would result from this flow chart. /3

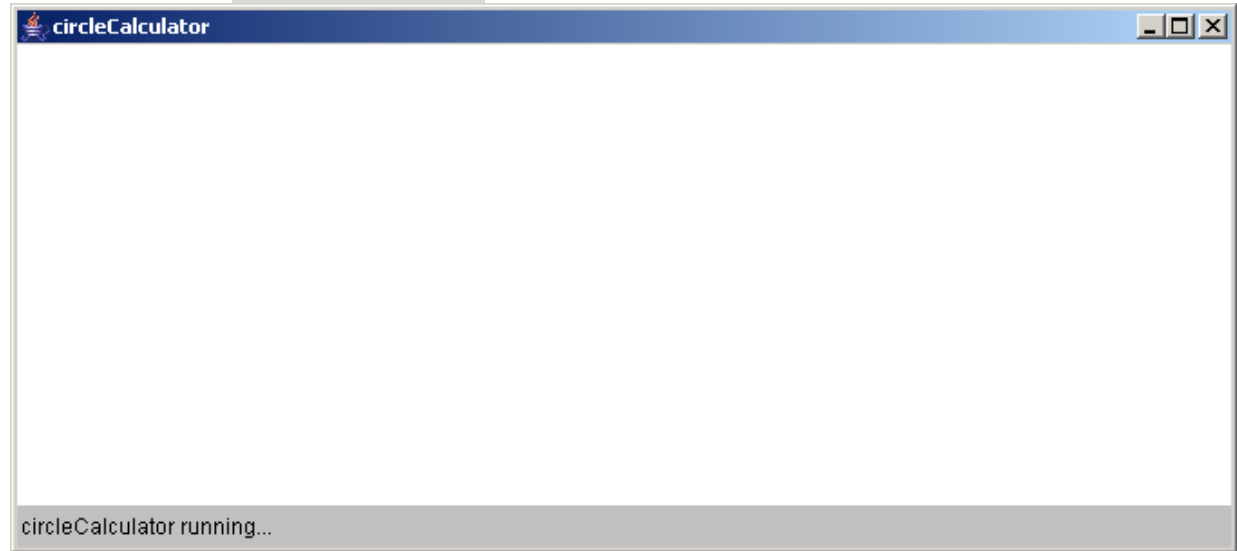


8. Look at the code below and circle the correct answer:

- T F a. You should save this applet as Calculator.java.
- T F b. There is a createImageIcon method in this program.
- T F c. An int is a widget.
- T F d. An accessor used in the program is getText.
- T F e. There are 7 widgets on the screen.
- T F f. setText is a mutator method.
- T F g. The output widget is globally declared.
- T F h. The widget whose text is changed in actionPerformed is showStatus.

9. Fill in the applet's screen using the code shown below.
Be careful to **label the colours**.

/12



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

public class circleCalculator extends Applet implements ActionListener
{
    JTextField radius;
    JLabel output;

    public void init ()
    {
        JButton b1 = new JButton ("Find Surface Area");
        b1.addActionListener (this);
        b1.setActionCommand ("calc1");
        b1.setBackground(Color.orange);

        JButton b2 = new JButton ("Find Volume");
        b2.addActionListener (this);
        b2.setActionCommand ("calc2");
        b2.setBackground(Color.orange);

        JLabel title = new JLabel ("Volume Calculations");
        title.setFont (new Font ("Ravie", Font.PLAIN, 20));
        title.setForeground (Color.red);

        JLabel n = new JLabel ("Enter the sphere's radius (cm):");

        radius = new JTextField (6);
        radius.setBackground (Color.yellow);

        output = new JLabel ("Please enter a radius and press a button.");
        output.setForeground (Color.blue);
```

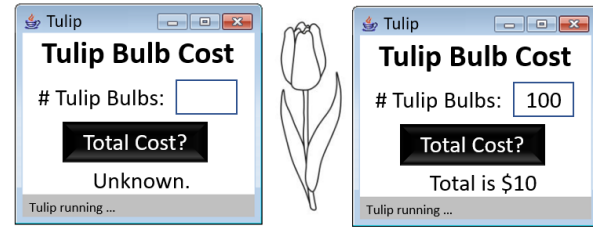
```
        add (title);
        add (n);
        add (radius);
        add (b1);
        add (b2);
        add (output);
    }

    public void actionPerformed (ActionEvent e)
    {
        int r = Integer.parseInt (radius.getText ());
        if (r <= 0)
            output.setText ("Enter a positive radius.");
        else if (r > 10000)
            output.setText ("Enter a smaller number.");
        else if (e.getActionCommand ().equals ("calc2"))
        {
            double v = 4.0 / 3.0 * Math.PI * r * r * r;
            output.setText ("The volume is " + v + ".");
        }
        else if (e.getActionCommand ().equals ("calc1"))
        {
            double sa = 4.0 * Math.PI * r * r;
            output.setText ("The surface area is " + sa + ".");
        }
        showStatus ("Thank you.");
    }
}
```

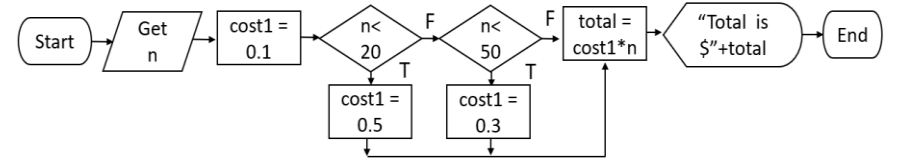
10. Code the init method of the Tulip applet. (see pictures to right) /10

```
import java.awt.*; import javax.swing.*; import java.applet.Applet; import java.awt.event.*;
public class Tulip extends Applet implements ActionListener
{
    JTextField num;
    //The other global variable:
    _____;
    public void init ()
    { resize (300, 100);
    //The first label: (Font is Arial, Font.BOLD and 30 pt)
    JLabel title = new _____ ("_____");
    title.setFont(new Font("_____",Font._____,____));
    //The prompt
    JLabel pmt = new _____ ("_____");
    //The textfield:
    num = new _____ (____);
    //The button: (Black background, white writing)
    JButton b = new _____ ("_____");
    b.setBackground(_____);
    b.setForeground(_____);
    b.addActionListener(_____);
    b.setActionCommand("_____");
    //The last label:
    _____;
    //add the widgets:
    add(_____);
    add(_____);
    add(_____);
    add(_____);
    add(_____);
} //init
```

11. Code the Tulip applet's actionPerformed method. /10



This flow chart explains the bulb pricing:



```
public void _____ (ActionEvent e)
{
    if(e.getActionCommand().equals("_____")){
        int n = Integer.parseInt(_____.getText());
        double _____ = 0.1;
        if (_____)
            cost1 = _____;
        else if (_____)
            cost1 = _____;
        double _____ = _____ * n;
        _____.setText("_____ "+_____);
    } //if
} //actionPerformed
} //Applet
```

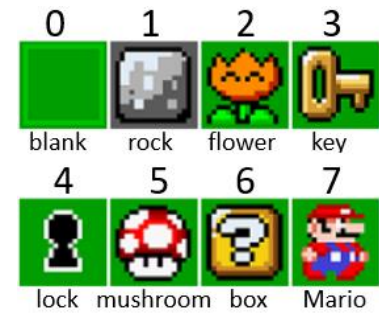
12. This is the first level in the Mario Flower Game.

/7



- (a) What is in land[0][0]?
- (b) What is in land[1][3]?
- (c) What is in land[3][1]?
- (d) What is in land[2][3]?
- (e) Where is the key (3)? land[...][...]
- (f) How many rows?
- (g) How many columns?

Picture information:



13. Write the code for the init method in the Mario Flower Game.

Mario's opening position is shown on the screen to the right. (Don't put Mario in the array)

/6

```
int land [][]= {{____, ____ , ____ , ____},
                {____, ____ , ____ , ____},
                {____, ____ , ____ , ____},
                {____, ____ , ____ , ____},
                {____, ____ , ____ , ____}};
```

```
int row = ____;    int col = ____;
int levelTotal = 4;
```

//Initial Mario position (on screen to right)

```
int x = ____;    int y = ____;
//Collected flowers & key variables
int flowers = 0;    boolean key = false;
```

```
JButton pics[]=new JButton[____*____];
Panel grid=new Panel(new GridLayout(____, ____));
int m=0;
for(int i=0; i<____; i++){
    for(int j=0; j<____; j++){
        pics[m]=new JButton (createImageIcon(____[i][j]+".jpg"));
        pics[m].addActionListener(this);
        pics[m].setActionCommand(m+"");
        grid.add(pics[m]);
        m++;
    }
}
add(grid);
```



14. Fill in the neighbours chart for the land array (above) in the Mario Flower Game.

/4

| | | |
|--------------------------|-------------------------------|---------------------------|
| | Up land[____][____] | |
| Left land[____][____] | Clicked Element land[x][y] | Right land[____][____] |
| | Down land[____][____] | |

15. The rules of the Mario Flower Game follow:

- Mario moves around the screen collecting flowers (2), getting a point each time.
- When Mario steps on a mushroom (5), he loses a flower point.
- When Mario goes to the key (3), he picks it up. The key variable becomes true.
- To win, Mario goes to the lock (4). He must have a key (true) and two flowers.

Code the down portion of actionPerformed.

/8

```

if (e.getActionCommand ().equals ("down")) {
    //edge guard for down
    if (_____ )
        showStatus ("Off the board!!");

    //down-one space has a rock
    else if (land [x + 1] [y] == _____)
        showStatus ("Rock!");

    else {
        showStatus ("Good move. ");

        //down-one has a flower
        if (land [x + 1] [y] == _____) {
            flowers++;
            //Change down-one to nothing
            land [_____] [_____] = _____;
        }

        //down-one space has a mushroom
        else if (land [x + 1] [y] == _____) {
            flowers--;
            //Change down-one space to nothing
            land [_____] [_____] = _____;
        }

        //down-one space has a key
        else if (land [_____] [_____] == _____) {
            key = _____;
            //Change the down space to nothing
            land [_____] [_____] = _____;
            showStatus ("You found the key!");
        }

        //down-one space has a lock, check for win
        else if (land [_____] [_____] == _____) {
            if (flowers >= _____ && key == _____)
                showStatus ("You win!!");
            else
                showStatus ("Need two flowers and a key to win");
        }

        score.setText ("Flower: "+_____+", Key: "+key);
        pics [x*col+y].setIcon (createImageIcon (land [x] [y]+".png"));
        x++;
        //change to Mario picture
        pics [x*col+y].setIcon (createImageIcon ("_____.png"));
    }
}

```



