



ICS4U **Sample** Final Exam

Computer Science

Friday January 24, 2024

Period 2 Exam Day, Room 129

Ms. Gorski

7 Question Pages

18 Questions

School Time Allotted: 8:30 – 11:00

Expected Time to Write Exam: 1.5 hours

Name (first and last): _____

Signature: _____

Total: / **140** %

Instructions:

- Students should answer all questions directly on the exam paper.
- Students may write in pencil or black pen or dark blue pen.
- Students will have access to the Objects Memory Aid as used on their Objects Test.

- 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. This is the original String:
String b = "BasketBALL";

0	1	2	3	4	5	6	7	8	9

/5

Match the code with the output.

10	a. System.out.println ((int) b.charAt (1));
2	b. System.out.println (b.charAt (0));
0	c. System.out.println (b.charAt (5));
97	d. System.out.println (b.compareTo ("hockey") < 0);
BASKETBALL	e. System.out.println (b.equals ("hockey"));
basketball	f. System.out.println (b.indexOf ('s'));
BasketBAMM	g. System.out.println (b.indexOf ('B'));
false	h. System.out.println (b.length ());
true	i. System.out.println (b.replace ('L', 'M'));
B	j. System.out.println (b.substring (0, 4));
t	k. System.out.println (b.substring (0, 1) + b.substring (4, 9));
Bask	l. System.out.println (b.toLowerCase ());
BetBAL	m. System.out.println (b.toUpperCase ());
BB	n. System.out.println (" " + b.charAt (0) + b.charAt (0));

2. This is the original String: String h = "hockey SKATES";

/10

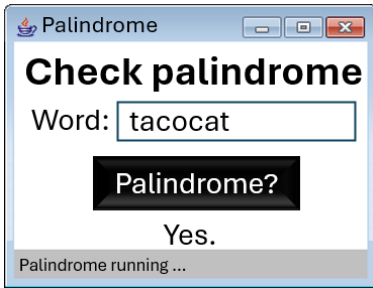
0	1	2	3	4	5	6	7	8	9	10	11	12

Write the code to produce the following output:

(a)	HOCKEY SKATES	System.out.println() ;
(b)	hockey skates	System.out.println() ;
(c)	A	System.out.println() ;
(d)	hockey PKATEP	System.out.println() ;
(e)	hockey	System.out.println() ;
(g)	13 //the length	System.out.println() ;
(h)	104 //ASCII of h	System.out.println() ;
(i)	2 //position of c	System.out.println() ;
(j)	S T ATE	System.out.println() ;
(k)	ok	System.out.println() ;

3. White Box and Black Box test this app.

/9



A palindrome is a word that reads the same backward as forward.

For example "racecar" or "nurses run".

Word	Black Box Test Case
Word	White Box Test Case & Line Number

```

1  public void actionPerformed (ActionEvent e){
2      String s = input.getText();
3      String rev = "";

4      for (int i = s.length () - 1 ; i >= 0 ; i--)
5          rev += s.charAt (i);

6      if (rev.equals (s))
7          output.setText ("Yes");
8      else
9          output.setText ("No");
10 }
    
```

4. Consider this recursive method, identify the following:

/9

```

public int mandlebrot (int n) {
    if (n <= 2)
        return 10;
    else
        return mandlebrot (n - 2) + 20;
}
    
```

(a)	The return type	
(b)	The method name	
(c)	The parameter name	
(d)	The parameter type	
(e)	Circle and label the base case and recursive case	<input type="checkbox"/>

Evaluate the following:

(f) mandlebrot(1).

(g) mandlebrot(4).

(h) mandlebrot(0).

5. Write a recursive method for the following sequence: 9 12 15 18 21 24 27 30 33 36 ...

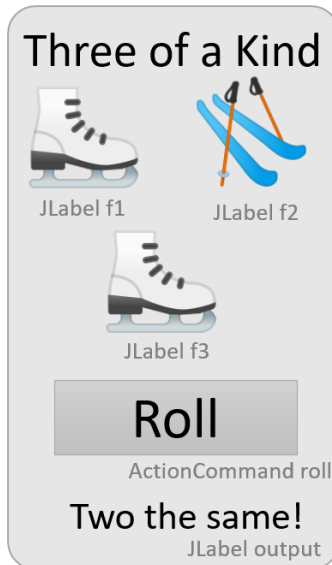
/6

For example, seq(1) returns 9 and seq(3) returns 15.

```

_____ ( _____ ) {
    method start word      return type (output)      method name (whatever)      parameter type      parameter name (input)
    
```

- (a) Circle the three sets of repeated code in the original actionPerformed.



The picture names, the corresponding random numbers.



```
public void actionPerformed(ActionEvent e) {
    int r1 = (int) Math.random()*4;
    if (r1 == 0)
        f1.setIcon(createImageIcon("ski.png"));
    else if (r1 == 1)
        f1.setIcon(createImageIcon("scarf.png"));
    else if (r1 == 2)
        f1.setIcon(createImageIcon("skate.png"));
    else
        f1.setIcon(createImageIcon("gloves.png"));
    int r2 = (int) Math.random()*4;
    if (r2 == 0)
        f2.setIcon(createImageIcon("ski.png"));
    else if (r2 == 1)
        f2.setIcon(createImageIcon("scarf.png"));
    else if (r2 == 2)
        f2.setIcon(createImageIcon("skate.png"));
    else
        f2.setIcon(createImageIcon("gloves.png"));
    int r3 = (int) Math.random()*4;
    if (r3 == 0)
        f3.setIcon(createImageIcon("ski.png"));
    else if (r3 == 1)
        f3.setIcon(createImageIcon("scarf.png"));
    else if (r3 == 2)
        f3.setIcon(createImageIcon("skate.png"));
    else
        f3.setIcon(createImageIcon("gloves.png"));
    calculateWin(r1, r2, r3);
    updateWinMessage();
}
```

- (b) Create a setPic method to replace the repeated code:

```
public void setpic(_____, int num) {
    if (_____ == _____)
        _____ .setIcon(createImageIcon("_____"));
    else if (_____ == _____)
        _____ .setIcon(createImageIcon("_____"));
    else if (_____ == _____)
        _____ .setIcon(createImageIcon("_____"));
    else
        _____ .setIcon(createImageIcon("_____"));
}
```

- (c) Call the setPic method for each picture in actionPerformed.

```
public void actionPerformed(ActionEvent e) {
    //random number for each
    int r1 = (int) Math.random()*4;
    int r2 = (int) Math.random()*4;
    int r3 = (int) Math.random()*4;
    //three method calls:
    _____ (_____, _____);
    _____ (_____, _____);
    _____ (_____, _____);
    calculateWin(r1, r2, r3);
    updateWinMessage();
}
```

7. What does ORATE stand for?

/5

--	--	--	--	--

8. In the first column, fill in the terms that match the description.

/8

	(a) Variables found inside an object's class.
	(b) A template for an object or type. Also contains a java program.
	(c) An object method type that returns values of instance variables.
	(d) An object method type that sets up dynamic memory.
	(e) An object method type that changes the values of instance variables.
	(f) Keeping an object's code self-contained and independent of other code. It relies only on itself.
	(g) A general term for data and the methods associated with that data.
	(h) Other programmers can use your objects without deep understanding of the specifics of how it is coded.

9. Circle the best data structure for each description.

/4

- Stack Queue (a) A FIFO structure.
 Stack Queue (b) A LIFO structure.
 Stack Queue (c) A pile of books model this data structure well.
 Stack Queue (d) The data structure that could model a waiting line.

10. Circle the **sorting** algorithm which is the best choice for each situation.

/4

- Bubble Quick Merge Selection (a) The list of names is sorted; you add one element to the front. You have extra memory.
 Bubble Quick Merge Selection (b) The array not randomized; it is very large. You just have enough memory to hold it.
 Bubble Quick Merge Selection (c) The char array is in reverse order, but it is only 62 elements long.
 Bubble Quick Merge Selection (d) You have lots of extra memory and the double array is in random order.

11. Circle the **search** which is the best choice for each situation.

/4

- Binary Linear (a) You have a list of heights, ordered from smallest to greatest.
 Binary Linear (b) You are looking up a word in a dictionary.
 Binary Linear (c) You look up a definition in a dictionary to find the word that goes with it.
 Binary Linear (d) You are looking up a student ID in a list with no apparent order.

12. Put these algorithm speeds in order.
 (1 is fastest, 6 is slowest)

13. What speed is each algorithm?

/10

- | | | |
|------------------------|-------------------|-------|
| ___ O(n) | (a) Swap | _____ |
| ___ O(n!) | (b) Quicksort | _____ |
| ___ O(log n) | (c) Binary Search | _____ |
| ___ Constant time | (d) Stack Pop | _____ |
| ___ O(n log n) | (e) BogoSort | _____ |
| ___ O(n ²) | (f) Linear Search | _____ |
| | (g) Queue Dequeue | _____ |

<pre> // Class line public _____ { // Instance Variables _____; _____; // Constructors public _____ () { _____ = _____; _____ = _____; } public _____ (_____, _____) { _____ = _____; _____ = _____; } // Mutator public _____ (_____) { _____ = _____; } public _____ (_____) { _____ = _____; } // Accessor public _____ (_____) { _____; } public _____ (_____) { _____; } // To String (Accessor) public _____ (_____) { _____ " _____ " + _____ + " " + _____; } // CompareTo, use last name public _____ (_____) { if(_____.compareTo(_____._____)>0) return _____; else if(_____.compareTo(_____._____)<0) return _____; else return _____; } </pre>	<pre> //Returns their initials public _____ (_____) { return _____ + _____; } //Equals (Facilitator) use first name public _____ (_____) { if(_____.equals(_____._____)) return _____; else return _____; } </pre> <p>(b) Adapt the Stack Class to make a Stack of Names.</p> <pre> public class _____Stack { private int count; private Object data[] = new Object [50]; public _____Stack () { count = 0; } public void push (Object addMe) { data [count] = addMe; count++; } public int size () { return count; } public boolean isFull () { return (count == 50); } public Object pop () {{ count--; return data [count]; } } public Object peek () { return data [count--]; } public boolean isEmpty () { return count == 0; } public void clear () { count = 0; } } </pre> <p>(c) Create a new Stack of names, add two names to it.</p> <pre> _____Stack s = new _____Stack(); Name n = new Name (_____, _____); s.push(____); Name m = new Name (_____, _____); s.push(____); </pre>
--	---

15. Using the pictures shown, fill in the code for this game of Farm Swap.

/5

The picture information:



The code:

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

int row = ____;

int col = ____;
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);
```

The grid information:



16. Fill in the neighbours chart for the center element.

/5

field[____][____]	field[____][____]	field[____][____]
field[____][____]	Clicked Element field[i][j]	field[____][____]
field[____][____]	field[____][____]	field[____][____]

17. In the game Farm Swap, you get extra points every time you get a frog (4) over an apple (2). In the screen shown, the user would get two extra points.

/8

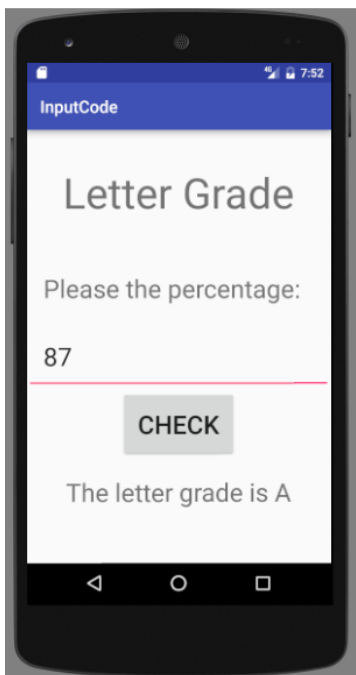
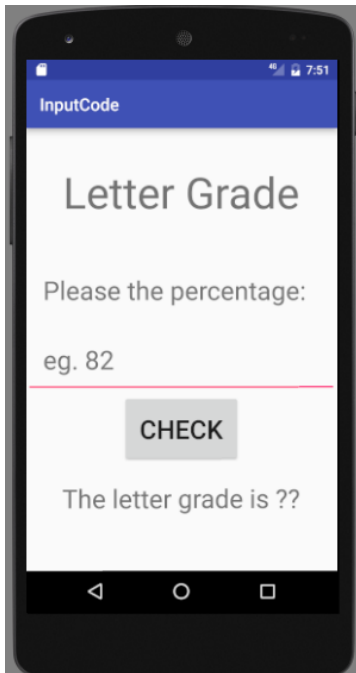
Code a method that returns the number of frogs over apples in a global array named field. The variables row and col track the dimensions for the array.



The method has no parameters but will return an integer.



```
public _____ ( _____ ) {
    _____ = _____;
    for ( _____ ) {
        for ( _____ ) {
            if ( _____ && field[____][____]==__ && field[____][____]==__ )
                _____;
        }
    }
    _____;
}
```



Percent	Grade
90+	A+
80-89	A
70-79	B
60-69	C
50-59	D
49-	F

```

<?xml version="1.0" encoding="utf-8"?>
< _____ xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:_____="match_parent"
    android:orientation="_____ ">
    < _____
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="40dp"
        android:text="_____ "
        android:textSize="50sp" />
    < _____
        android:layout_width="wrap_content"
        android:layout_height="_____ "
        android:padding="20dp"
        android:text="Please the percentage:"
        android:textSize="30sp" />
    < _____
        android:id="@+id/input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="eg. 82"
        android:padding="20dp"
        android:textSize="30sp" />
    < _____
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:onClick="_____ "
        android:padding="20dp"
        android:text="_____ "
        android:textSize="30sp" />
    < _____
        android:_____="@+id/output"
        android:layout_width="_____ "
        android:layout_height="_____ "
        android:layout_gravity="center"
        android:padding="20dp"
        android:text="_____ "
        android:textSize="30sp" />
</ _____ >

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void clicked(_____ ) {
        _____ input = (EditText) findViewById(R.id._____ );
        TextView output = (_____ ) findViewById(R.id._____ );
        int num = Integer.parseInt(_____ .getText().toString());
        if(_____ )
            _____ ("_____ ");
        else if(_____ )
            _____ ("_____ ");
        else if(_____ )
            _____ ("_____ ");
        else if(_____ )
            _____ ("_____ ");
        else if(_____ )
            _____ ("_____ ");
        else
            _____ ("_____ ");
    }
}
    
```