Part I. Multiple Choice Questions (2 points each):

1. What happens when you try to assign a value larger than the maximum possible integer to an int variable?

   (a) The program terminates

   (b) You get an overflow error

   (c) The value becomes a negative value *****

   (d) The value becomes zero

2. Which of the following patterns is produced when the following code snippet is executed?

```
for(int i = -2; i <= 2; i++){
    for(int j = -2; j <= 2; j++){
        if(i < j)
            System.out.print("*");
        else
            System.out.print(".");
    }
    System.out.println();
}
```

(a)
```
*....
**...
***..
****.
*****
```

(b)
```
*....
**...
***..
****.
*****
```

(c)
```
.....
....*
...**
..***
.****
```

(d)
```
.****     *****
..***
...**
....*
.....
```

3. What keyword is used to specify that a data member is a class data member (shared among all instances of that class)?

   (a) final
   (b) shared
   (c) public
   (d) static *****

4. Which of the following is a characteristic of an overloaded method?

   (a) it must have a fixed number of parameters
   (b) it could never be a constructor
   (c) it shares a name with another method in the same class *****
   (d) it always returns void

5. What is the result of x when the following loop exits?

```
int x = 0;
for(int i = 0; i < 10; i++){
    if(i == 5){
        continue;
    }
    x++;
}
```

   (a) 5
   (b) 10
   (c) 9 *****
   (d) 6

6. Which of the following is a true statement regarding a constructor?

   (a) it has no return type *****
   (b) its return type changes based on how you write it
   (c) its return type is void
   (d) its return type is int

7. what is the result of count after the following loop exits?

```
int count = 10;
do{
    count--;
}while(count >= 10);
```

   (a) 8
   (b) 11
   (c) 9 *****
   (d) 10

8. Which of the following is true about exceptions in Java?

```
 I.  A RuntimeException does not need to be caught by a try/catch block.
II.  A NumberFormatException must be caught by a try/catch block.
III. A NullPointerException does not need to be caught by a try/catch block.
```

(a) I only

(b) I and III *****

(c) II and III

(d) III only

9. Given the program below, what will be the values of asum and bsum?

```
public class MyArray
{
    private int [ ] X={3, 4, 5};
    public MyArray( )
    {
    }
    public int value( )
    {
        return(X[0]);
    }
    public int value(int k)
    {
        return(X[k]);
    }
    public static void main (String[ ] args)
    {
        MyArray [ ] Y = new MyArray[10];
        int asum = 0, bsum = 0;
        for(int i=0; i<10; i++)
        {
            Y[i]=new MyArray( );
            asum = asum + Y[i].value( );
            for(int j=0; j<3; j++)
                bsum = bsum + Y[i].value(j);
        }
        System.out.println("asum: "+ asum + " bsum: " + bsum);
    }
}
```

(a) asum: 120 bsum: 120

(b) asum: 30 bsum: 120 *****

(c) asum: 120 bsum: 30

(d) asum: 30 bsum: 30

10. Which of the following is true about try/catch blocks in Java?

```
 I.  All try/catch blocks must have a finally block.
II.  A try/catch block is limited to two or less catch blocks.
III. Barring the Java virtual machine from exiting, the finally
     block will always be executed.
```

(a) I and III

(b) I only

(c) III only *****

(d) I and II

11. What will the word “apple” become after being encrypted 101 times using the following encryption table?

```
a -> e
e -> l
l -> p
p -> a
```

(a) eaapl *****

(b) pllea

(c) leeap

(d) apple

12. What is the output of the following code?

```
try {
    int num  = Integer.parseInt("four thousand two hundred and ninety-five");
    System.out.println("Your number is: " + num + ".");
} catch (NumberFormatException n) {
    System.out.println("You don’t have a number.");
} catch (Exception e) {
    System.out.println("Something went terribly terribly wrong.");
} finally {
    System.out.println("Number parsed successfully!");
}
```

(a) `You don’t have a number.` *****
`Number parsed successfully!`

(b) `Something went terribly terribly wrong.`
`Number parsed successfully!`

(c) `Your number is: 4295.`
`Number parsed successfully!`

(d) `You don’t have a number.`

13. Which subclass of Throwable is an exception checked at compile time?

    (a) NullPointerException
    (b) RuntimeException
    (c) IOException *****
    (d) ArrayIndexOutOfBoundsException

14. What is displayed by the following code?

```
String result = "";
int[] arrr = { 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 };
for ( int index = 0; index < 10; ++index )
    result += (arrr[ index*index ] + " ");
System.out.println( result );
```

    (a) 9 8 7 6 5 4 3 2 1 0
    (b) 0 1 4 9 16 25 36 49 64 81
    (c) 9 8 5 0
    (d) an ArrayIndexOutOfBoundsException *****

15. Which layout manager places components in one of five regions: north, south, east, west, and center?

    (a) AbsoluteLayout
    (b) GridLayout
    (c) BorderLayout *****
    (d) FlowLayout

16. A frame's _____ designates the area of the frame excluding the title, menu bar, and the border.

    (a) content pane *****
    (b) layout manager
    (c) event listener
    (d) action listener

17. Clicking a mouse button will always generate which event?

    (a) MouseButtonEvent
    (b) ActionEvent
    (c) MouseClickEvent
    (d) MouseEvent *****

18. What is displayed by the following code?

```
void shiverTimbers( int[] yarrr )
{
    int[] local = yarrr;
    for ( int jolly = 0; jolly < local.length/2; ++jolly )
    {
        int blackbeard = local[jolly];
        local[jolly] = local[local.length - 1 - jolly];
        local[local.length - 1 - jolly] = blackbeard;
    }
}
...
int[] roger = { 0, 78, 74, 99, 69, 9 };
shiverTimbers( roger );
System.out.print( roger[0] );
System.out.println( ", " + roger[5] );
```

(a) 0, 9

(b) 74, 0

(c) 9, 0 *****

(d) 0, 74

19. What is displayed by the following code?

```
ArrayList plank = new ArrayList( 5 );
int sum = 0;
for ( int peg = 0; peg < 20; ++peg )
    plank.add( peg, new Integer(peg) );
for ( int iPatch = 0; iPatch < 20; iPatch+=2 )
    sum += ((Integer)plank.get( iPatch )).intValue();
System.out.println( sum );
```

(a) an InvalidArgumentException

(b) 210

(c) 90 *****

(d) an ArrayIndexOutOfBoundsException

20. What is the value of arrr after the following?

```
int[][] arrr = { {0}, {0,1}, {0,1,2}, {0,1,2,3} };
for ( int i = 0; i < arrr.length; ++i )
{
    for ( int j = 0; j < arrr[i].length; ++j )
    {
        arrr[i][j] = arrr[3-j][3-i];
    }
}
```

(a) 0, 0, 1, 0, 1, 1, 0, 0, 0, 0

(b) 3, 2, 2, 1, 1, 2, 0, 1, 2, 3 *****

(c) 0, 1, 1, 2, 2, 1, 3, 2, 1, 0

(d) 0, 0, 1, 0, 1, 2, 0, 1, 2, 3

The version of your test is **A**. Please **FILL IN CIRCLE (A) for the TEST FORM field on the BUBBLE SHEET** directly under the DATE field and turn in your exam booklet and answer sheet to the stack labeled (A) in the front of the classroom. Thank you.

This page is left blank intentionally.

Part II. Programming Questions (60 points total):

1. (20 points) You are creating a High-Low game. The first player will enter the “magic number”. Your program must accept a “magic number” greater than zero and less than 100. You should continue prompting for the “magic number” until a valid number is entered.

   You should then prompt the second player for their guess. You should print out “Higher” if the magic number is higher than the guess and you should print out “Lower” if the magic number is less than the guess. Continue prompting the user until they have guessed the number. When the magic number has been guessed you should print out a corresponding message and then end the program.

   You should use the JOptionPane to prompt for input and display output. Remember it will return a string you will have to parse. You can assume all input will be integers. You are not required to write methods. All of your code can be in the main function.

Solution for programming question 1:

```
import javax.swing.*;

public class prob1 {

    public static void main(String args[]){

        int magicNum;
        String numString;
        int num = -1;

        do{
            numString = JOptionPane.showInputDialog(null,
                                                "Input the Magic Number");
            magicNum = Integer.parseInt(numString);

        }while(magicNum < 0 || magicNum > 100);

        while(num != magicNum){
            numString = JOptionPane.showInputDialog(null, "Input Guess");
            num = Integer.parseInt(numString);

            if(num < magicNum){
                JOptionPane.showMessageDialog(null, "Higher");
            }
            else if(num > magicNum){
                JOptionPane.showMessageDialog(null, "Lower");
            }
        }
        JOptionPane.showMessageDialog(null, "you've guessed the number");
    }
}
```

2. (20 points) You are given a partially complete SimpleGUI class which pops up a GUI with one text field, two buttons ("ok" and "cancel"), and a feedback label. You must augment this class to make it respond to button events.

   Implement the following behavior:

   - ok button pushed:
     - Set the feedback label to the text in the textField.
     - Clear the textField.
   - cancel button pushed:
     - Set the feedback label to "cancel pushed".
     - Clear the textField

   You must add the necessary code to the SimpleGUI constructor in order for this class to receive action events. You must add the necessary code to implement the described behavior. **If you need to alter the class declaration, you should be sure to do that as well.**

   SimpleGUI template is given on the following pages.

This page is left blank intentionally.

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

public class SimpleGUI extends JFrame
{
   private Container contentPane;
   private JTextField textField;
   private JButton ok, cancel;
   private JLabel feedback;
   private JPanel panel;

   public static void main( String[] args ) {
      SimpleGUI gui = new SimpleGUI();
      gui.pack();
      gui.setVisible( true );
   }

   public SimpleGUI() {

      contentPane = getContentPane();
      contentPane.setLayout( new GridLayout( 3, 1 ) );

      textField = new JTextField();
      textField.setColumns( 22 );
      contentPane.add( textField );

      panel = new JPanel( new FlowLayout() );
      ok = new JButton( "OK" );
      cancel = new JButton( "Cancel" );
      panel.add( ok );
      panel.add( cancel );
      contentPane.add( panel );

      feedback = new JLabel();
      contentPane.add( feedback );

      // TODO Part1
      // Insert code here to register this frame to receive action events

      setDefaultCloseOperation( EXIT_ON_CLOSE );
   }

  // TODO Part2
  // Insert code here to implement the described behavior

}
```

Solution for programming question 2: Please note you do not need to copy any of the given code. Just label your code with TODO Part1 and TODO Part2. If you need to do any other code modification of the given template, please label it TODO Part3.

TODO Part1

```
    ok.addActionListener( this );
    cancel.addActionListener( this );
```

TODO Part2

```
  public void actionPerformed( ActionEvent event ) {

    if( event.getSource() == ok ) {

      feedback.setText( textField.getText() );
      textField.setText( "" );

    } else if( event.getSource() == cancel ) {

      textField.setText( "" );
      feedback.setText( "cancel pushed" );

    }

  }
```

TODO Part3

```
public class SimpleGUI extends JFrame \textbf{implements ActionListener}
```

This page is left blank intentionally.

This page is left blank intentionally.

3. (20 points) Write a program that asks the user to enter 300 integers between 0 and 2000 inclusive. After the user enters these numbers, the statistical mode should be displayed. The statistical mode is the most frequently occurring number. If there is a tie for the mode, ”no mode” should be displayed. You need not perform error checking.

   Example:

```
Enter an integer in [0,2000]:
3
Enter an integer in [0,2000]:
3
Enter an integer in [0,2000]:
3
Enter an integer in [0,2000]:
5
Enter an integer in [0,2000]:
8
...

Mode: no mode
```

   Hint: You should consider using an array of integers to keep a running total of the number of times a particular number has been entered. e.g. count[180] can contain the total number of times the user entered the number 180.

Solution for programming question 3:

```
import java.util.Scanner;

public class prob3
{
    public static void main( String[] args )
    {
        final int NUM_TO_ENTER = 300;
        Scanner keyboard = new Scanner( System.in );
        int[] count = new int[2001];

        for ( int num = 0; num < NUM_TO_ENTER; ++ num )
        {
            System.out.println( "Enter an integer in [0,2000]: " );
            int newNum = keyboard.nextInt();
            ++count[newNum];
        }

        boolean hasMode = true;
        int mode = 0, modeCount = 0;
        for ( int i = 0; i < count.length; ++i )
        {
            if ( count[i] > modeCount )
            {
                mode = i;
                modeCount = count[i];
                hasMode = true;
            }
            else if ( count[i] == modeCount )
                hasMode = false;
        }

        System.out.println( "Mode: " + (hasMode ? mode : "no mode") );
    }
}
```