ITSE 2417 (DL)- Fall 2016
Quiz 2- Chapters 8-13
Total Points: 60

Due: Saturday, October 29th @ 11:59PM. Look at Syllabus/ICR on late work.

Directions: For Questions 1-26, clearly mark answers on a separate word (or notepad) document. See sample file/directions provided by your professor and submit to the appropriate location on the MyTCC (BlackBoard) site.

– Assume all variables are properly declared- unless otherwise mentioned.

Multiple Choice. Mark the one best answer for each question. **(2 pts. each)**

1. Which method can be used to create an input object for file `temp.txt`?
   A. `new Scanner("temp.txt")`
   B. `new Scanner(temp.txt)`
   C. `new Scanner(new File("temp.txt"))`
   D. `new Scanner(File("temp.txt"))`

2. ____ exceptions are the type that programmers should anticipate and from which programs should be able to recover.
   A. Unchecked
   B. Runtime
   C. Checked
   D. Thrown

3. Which of the following lines would correctly fill **`<blank 1>`** to complete the `S toString()` method?

```
public class R
{
   private int one;
   public R(){ one = 0; }
   public String toString(){
      return "" + one;
   }
}

public class S extends R
{
   private int two;
   public S() { two = 1; }
   public String toString(){
      <blank 1>
   }
}
```

   A. `return super.toString() + " "  + two;`
   B. `return " "  + two;`
   C. `return super.toString();`
   D. `return toString() + " "  + two;`

4. Which methods must be implemented in any class that inherits from the class `Stuff`?

```
public abstract class Stuff
{
   private int x;

   public Stuff()
   {
      x = 5;
   }

   public Stuff(int newX)
   {
      x = newX;
   }

   public abstract void doStuff();
   public abstract int doubleIt();
}
```

A.
```
Stuff()
Stuff(int newX)
doStuff()
doubleIt()
```

B.
```
Stuff()
doStuff()
doubleIt()
```

C.
```
doStuff()
```

D.
```
doStuff()
doubleIt()
```

5. What is displayed on the console when running the following program?

```
class Test {
  public static void main(String[] args) {
    try {
      method();
      System.out.println("After the method call");
    }
    catch (RuntimeException ex) {
      System.out.println("RuntimeException");
    }
    catch (Exception ex) {
      System.out.println("Exception");
    }
  }

  static void method() throws Exception {
    try {
      String s = "5.6";
      Integer.parseInt(s); // Cause a NumberFormatException

      int i = 0;
      int y = 2 / i;
      System.out.println("Welcome to Java");
    }
    catch (NumberFormatException ex) {
      System.out.println("NumberFormatException");
      throw ex;
    }
    catch (RuntimeException ex) {
      System.out.println("RuntimeException");
    }
  }
}
```

A. The program displays `NumberFormatException` twice.
B. The program displays `NumberFormatException` followed by `After the method call`.
C. The program displays `NumberFormatException` followed by `RuntimeException`.
D. The program has a compilation error.

6. Which of the following statements about the reference `super` is true?
A. It must be used every time a method from the superclass is called.
B. It must be the last statement of the subclass constructor.
C. It must be the first statement of the subclass constructor.
D. It can only be used once in a program.

7. What is output by the following code segment?

```
String a = "Java";
String b = "is";
String c = "Great";
String d = c;
System.out.println(d.equals(c));
```

A. `0`
B. `-1`
C. `false`
D. `true`

8. An interface can extend how many other classes?
A. 0
B. 1
C. 2
D. as many as you want

9. Which of the following could not fill **`<blank 1>`**?

```
public interface Testable
{
   <blank 1>
}
```

A. `private void setScore(double s);`
B. `void setScore(double s);`
C. `abstract void setScore(double s);`
D. A and B only

10. What is output by the code below?

```
int[][] m = new int[7][7];

System.out.println(m[3][4]);
```

A. `null`
B. `5`
C. `9`
D. `0`

11. `String`s and other objects that can't be changed are known as ____.
A. string constants
B. accessor methods
C. immutable
D. garbage

12. What is output by the code below?

```
public class A
{
  private int one;
  public A(int o){
    one = o;
  }
  public void setOne(int o){
    one = o;
  }
  public int getOne(){
    return one;
  }
}

public class B
{
  public void changeOne(A param){
    param = new A(0);
    param.setOne(7);
  }
  public void changeTwo(A param){
    param.setOne(4);
  }
  public void changeThree(A param){
    param.setOne(9);
    param = new A(0);
  }
}

//code in the main of another class
B test = new B();
A theA = new A(2);
test.changeThree(theA);
System.out.println(theA.getOne());
```

A. `2`
B. `0`
C. `7`
D. `9`

13. The reference to an object that is passed to any object's nonstatic class method is called the ____.
    A. magic number
    B. `this` reference
    C. literal constant
    D. reference

14. Assume `int[][] x = {{1, 2}, {3, 4, 5}, {5, 6, 5, 9}};`, what are `x[0].length`, `x[1].length`, and `x[2].length`?

    A. 2, 3, and 3
    B. 2, 3, and 4
    C. 3, 3, and 3
    D. 2, 2, and 2

15. What is output by the code below?

```
String r = "racecar";
System.out.println(r.lastIndexOf('a'));
```

A. -1
B. 1
C. 4
D. 5

16. Which of the following statements are correct?

I

```
File file = new File("input.txt");
try (Scanner input = new Scanner(file)) {
  String line = input.nextLine();
}
```

II

```
try (File file = new File("input.txt");
     Scanner input = new Scanner(file);) {
  String line = input.nextLine();
}
```

III

```
File file;
try (file = new File("input.txt");
     Scanner input = new Scanner(file);) {
  String line = input.nextLine();
}
```

IV

```
File file;
Scanner input;
try (file = new File("input.txt");
     input = new Scanner(file);) {
  String line = input.nextLine();
}
```

A. I
B. II
C. III
D. IV

17. Any new class you create from an existing class is called a(n) ____.
A. base class
B. superclass
C. derived class
D. extended class

18. A child having a different version of a parent's method is an example of what?
A. method overloading
B. method overriding
C. method rewriting
D. parent extention

19. Consider the following instance variables and methods that appear in a class representing matrix information.

```
private int[][] mat;

public int totalRow(int r)
{
   /* code */
}
```

The `totalRow` method will sum up all values on the provided row `r`.

Consider the following proposed implementations of the `totalRow` method.

I.

```
int sum = 0;
for(int i = 0; i < mat[r].length; i++)
   sum = sum + mat[r][i];
return sum;
```

II.

```
int sum = 0;
for(int i = 0; i < mat.length; i++)
   sum = sum + mat[r][i];
return sum;
```

III.

```
int sum = 0;
for(int i = 0; i < mat[r].length; i++)
   sum = sum + mat[i][r];
return sum;
```

Which of the proposed implementations will correclty implement method `totalRow`?

A. I only
B. II only
C. III only
D. I and II only

20. Given the following code:

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

   public Point()
   {
      x = 0;
      y = 0;
   }

   public Point(int nx, int ny)
   {
      x = nx;
      y = ny;
   }
}
public class PointThree extends Point
{
   private int z;

   public PointThree()
   {   /* Implementation not shown */   }

   public PointThree(int nx, int ny, int nz)
   {   /* Implementation not shown */   }

}
```

Which of the following correctly implements the default `PointThree` construtor?

A.
```
super();
z = 0;
```

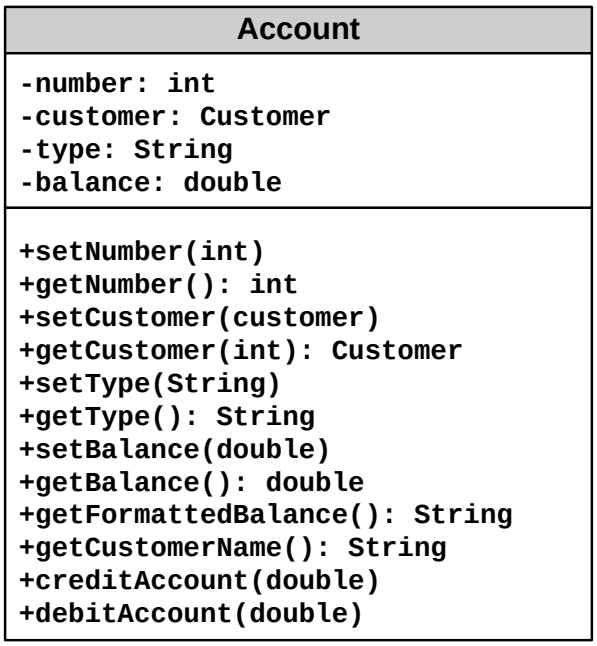
B.
```
z = 0;
super();
```

C.
```
x = 0;
y = 0;
z = 0;
```

D.
```
super();
y = 0;
z = 0;
```

Short Answer. Clearly mark answers as directed. Partial Credit will be given. (**10 @ 2 each)**

Use the following UML Class Diagram to answer Questions 21-25:



21. Code a constructor for the `Account` class that assigns values to the `number, type,` and `balance` instance variables based on the values that are passed to it, and assigns a value to the `customer` instance variable by calling the `getCustomer` method. The parameters of this constructor should be given the same names as the instance variables. The `getCustomer` method should accept the number variable as a parameter and return a `Customer` object. Write this code as concisely as possible.

22. Write the code for the `setType` method. This method should simply set the value of the type instance variable to the value that's passed to it.

23. Code a statement that sets the value of the type instance variable by calling the `setType` method. Assume that an instance of the `Account` class has already been created and stored in a variable named `account` and that a variable named `type` that stores the account type has already been declared and assigned a value in the calling class.

24. Declare a static field named `tranCount` that can be used to track the number of debits and credits that are posted to all of the account objects created from the `Account` class. Assume that this field will be updated only from the `creditAccount` and `debitAccount` methods of the class.

25. Code a method named `getTranCount` that can be used by other classes to get the value of the

26. Write a Java program that throws and catches an `ArithmeticException` when you attempt to take the square root of a negative value. Prompt the user for an input value and try the `Math.sqrt()` method on it. The program either displays the square root or catches the throw `Exception` and displays the appropriate message.

Sample Runs:

```
Enter a number: 12
Result is: 3.4641016151377544

Enter a number: -23
Can't take square root of negative number
```

Fill in the missing parts of the program below to solve the problem as stated above. Do not add any additional lines of code. **(10 @ 2 each)**

```
//SqrtException.java
import java.util.Scanner;

_____________________                                          //1
{
   public static void main(String[] args) throws ArithmeticException
   {

      Scanner in = new Scanner(System.in);
      double num, result;

      System.out.print("Enter a number: ");
      num = in.nextDouble();
      _______________                                          //2
      {
         if(num < 0)
            ___________________                                //3
         result = Math.sqrt(num);
         ____________________                                  //4
      }
      ____________________                                     //5
      {
         System.out.println("Can't take square root of negative number");
      }
   }
}
```

Extra Credit: Implement the following program. Follows same program guidelines and graded on the same scale as program sets. Submit only your `.java` file- no test runs required. Partial credit given. (**10 points)**

Write a Java program that reads in a list of words from a file (`words.txt`) and prints only those words ending in the letters "`ED`". Output the list of words three words per line. Assume only one word per line in the file. Let the user input the file name from the keyboard. Output should be user friendly.

Name the program: `OutputEDWordsXX.java`, where `XX` are your initials.