CSCI 6010 Introduction to Computer Science Fundamentals, Spring 2016
Mid-Term Exam 7pm March 4th – midnight March 6th

I pledge that this test/assignment has been completed in compliance with the Graduate Honor Code and that I have neither given nor received any unauthorized aid on this test/assignment.

Name (Print): Kedar Ravindra Haldankar
Signed: _____

Mid-term solutions

1	A
2	C
3	A, E
4	D
5	C
6	C
7	D
8	25
9	just right
10	Income is greater than 3000
11	A, B, C, D
12	A, C
13	3 0
14	120 elements , x.length=12, x[2].length=5, x[0][0].length=2
15	C, D
16	Error: Paranthesis are not placed proper in last system.out.println. **If corrected**, output will be : 3 -1
17	(a) 0 1 (b) 0

18	(a) 2
	2 4
	2 4 8
	2 4 8 16
	2 4 8 16 32
	2 4 8 16 32 64
	
	(b) `Before the call, variable times is 3`
	`n = 3`
	`Welcome to Java!`
	`n = 2`
	`Welcome to Java!`
	`n = 1`
	`Welcome to Java!`
	`After the call, variable times is 3`

19	Exception: FileNotFoundException has been caught twice.

20	<pre> 1
	         2 1
	       3 2 1
	     4 3 2 1
	   5 4 3 2 1
	6 5 4 3 2 1</pre>

21	Error: character passed instead of string, **if passed "a"**, output will be aaaa
22	B
23	C, D, E, F
24	1 8
25	Error: Constructor accepts string argument which is not provided.
26	False
27	A, B, C, D, F, I
28	E
29	(a) A's no-arg constructor is invoked (b) Error: No arguments are passed / `constructor A in class A cannot be applied to type B.`
30	In constructor of Circle, and class B, Statements should be changed to: **this.radius=radius;** **super(radius)** **this.length=length;** respectively. Also, in getArea() method in class B, return statement should be : **super.getArea() * this.length.** If none of these is done then it will throw an error as incompatible types.
31	(a) Person Student (b) Person Person
32	

	`public static long FibonacciIteration(long n)`
	`    {`
	`        if (n == 0) return 0;`
	`        if (n == 1) return 1;`
	`        long x=0;`
	`        long y=1;`
	`        long z=0;`
	
	`        for (int i = 0; i < n-1; i++)`
	`        {`
	`            z = x + y;`
	`            x = y;`
	`            y = z;`
	`        }`
	
	`        return z;`
	
	`    }`

33	(a) Error: method add in class test cannot be applied to given types; add(stack1, stack2); ^ required: Stack<T>,Stack<T> found: Stack<String>,Stack<Object> reason: inferred type does not conform to equality constraint(s) inferred: Object equality constraints(s): Object,String where T is a type-variable: T extends Object declared in method <T>add(Stack<T>,Stack<T>) **(b) Original output does not change:** Sun 2 Java
35	Please check on next page:

```
public class Hexagon extends GeometricObject implements Cloneable,
    Comparable<Hexagon> {
  private double side;

  /** Construct a Hexagon with the specified side */
  public Hexagon(double side) {
    // Implement it
    this.side=side;

  }

  @Override
  public double getArea() {
     // Implement it (area = 3*√3/2*side*side)
     double area=3*side*side*(Math.sqrt(3)/2);
     return area;

  }

  @Override
  public double getPerimeter() {
    // Implement it
    double perimeter=6*side;
    return perimeter;

  }

  @Override
  public int compareTo(Hexagon obj) {
    // Implement it (compare two Hexagons based on their sides)
          double secondPerimeter=obj.getPerimeter();
          double secondSide= secondPerimeter/6;
          double difference= this.side-secondSide;
          if(difference==0.00)
           return 0;
          else{
           if (difference > 0.00)
           return 1;
           else{

           return -1;
           }
          }


  }

  @Override
  public Object clone() {
    // Implement it
```

```
        try{
              return super.clone();
        }
        catch(Exception Ex)
        {
              return null;
        }


    }
}
```

36

(a) Here, R={<a,b>|iff a/b=integer} where a∈S and b∈S, where S is set of all nonzero rational numbers.

In order to be an equivalence relation, the relation must be **reflexive**, **symmetric** and **transitive.**

i. Reflexivity test:
consider any random number from R. Let's just take 4.
In order to be reflexive aRa should be valid for all a∈R
Here, 4R4 = 4/4=1 is an integer. And it is true for all the elements of set S. Hence **Relation R is reflexive.**

ii. Symmetric test:
Consider any two numbers a and b from set R,
In order to be symmetric if aRb is valid, then bRa should also be valid.
Let's take 12 and 6. In this case we have 12R6= 12/6=2 is an integer. But 6R12= 6/12= 1/2 is not an integer. Hence **Relation R is NOT symmetric.**

From these two tests we can prove that **this is not an equivalence relation.**

(b) Here, R={<a,b>|iff |a-b|<=2} where a∈S and b∈S, where S is set of all rational numbers.

In order to be an equivalence relation, the relation must be **reflexive**, **symmetric** and **transitive.**

i. Reflexivity test:
consider any random number from R. Let's just take 4.
In order to be reflexive aRa should be valid for all a∈R
Here, 4R4 = |4-4|<=2 = 0<=2 is valid. And it is

	true for all the elements of set S. Hence **Relation R is reflexive.** ii. Symmetric test: Consider any two numbers a and b from set R, In order to be symmetric if aRb is valid, then bRa should also be valid. Let's take 2 and 3. In this case we have 2R3= |2-3|<=2 is valid. Also, 3R2= |3-2|= 1 is valid. Hence **Relation R is symmetric.** iii. Transitivity test: Consider any three numbers a, b and c from set R, In order to be transitive, if aRb and bRc then aRc must be valid Let's take 2, 4, 6. We have 2R4= |2-4| = 2 <=2 is valid 4R6= |4-6| = 2 <=2 is valid 2R6=|2-6| = 4 <= 2 in **NOT valid** Hence **R is not transitive.** From these three tests we can prove that **this is not an equivalence relation.**
37	Consider n is not prime when 2^n-1 is prime. Let's say 2^n-1= 5, Hence, 2^n=6 $2^{(n-1)}$ * 2= 2* 3 $2^{(n-1)}$=3 according to the hint a=n, b=1. $(2^n-1)*(2^{n(1-1)})$=3 as $2^{n(1-1)}$=1, 2^n-1=3 2^n=4 **n=2, here n is prime** **hence our assumption is wrong and it's proved that 2^n-1 is prime, then n is prime.**
38	$n!$, e^n, $(3/2)^n$, $n^{\lg \lg n}$, $2^{\lg n}$, $(\sqrt{2})^{\lg n}$, $\lg^2 n$, $\ln n$, $n^{1/\lg n}$
39	This corresponds to the harmonic series H_n. i.e $T(n)=H_n$ $H_{n=}=1/1+1/2+1/3+\ldots..+1/n$ $T(1)=H_1=1$ $T(n-1)= H_{n-1}$ So given expression becomes $T(n)=H_{n-1} + 1/n = H_n$

	As $H_n = \Theta(\lg n)$ **$\underline{T(n) = \Theta(\lg n)}$**
40	$T(n) = T(\sqrt{n}) + 1$ $T(n) = T(n^{1/2}) + \Theta 1$ $T(n) = T(n^{1/4}) + \Theta 1 + \Theta 1$ This will continue k times, until $n^{1/2\wedge k} <= 1$ $2^k = \lg n$ $k = \lg \lg n$ **$\underline{T(n) = k * \Theta(n) = \Theta(\lg \lg n)}$**
41	(a) Let us assume: D=Maximum number of elements in the list E=The Size of the data element(In this case it's 4 bytes) P= The size of pointer (in this case 4 bytes) The case in which the array-based list is more space efficient than the linked list: $n > D*E/P+E$ $n > (D*4)/(4+4)$ $n > (D*4)/8$ **$\underline{n > D/2}$** **Hence, when 'n' i.e. number of elements currently in the list are more than half of 'D' i.e. maximum number of elements in the list, the array-based list is more space efficient than the linked list** (b) Let us assume: D=Maximum number of elements in the list E=The Size of the data element(In this case it's 8 bytes) P= The size of pointer (in this case 4 bytes) The case in which the array-based list is more space efficient than the linked list: $n > D*E/P+E$ $n > (D*8)/(8+4)$ $n > (D*8)/12$ **$\underline{n > D*2/3}$** **Hence, when 'n' i.e. number of elements currently in the list are more than two third of 'D' i.e. maximum number of elements in the list, the array-based list is more space efficient than the linked list**

42	(a) $T(\alpha n)+T((1-\alpha)n)+cn$ $=d\alpha n \log(\alpha n) + d(1 - \alpha)n \log((1 - \alpha)n) + cn$ $d\alpha n\log\alpha+d\alpha n\log n+d(1-\alpha)n\log(1-\alpha)+d(1-\alpha)n\log n+cn$ $=dn\log n+dn(\alpha\log\alpha+(1-alpha)\log(1-alpha))+cn$ $=dn \log n.$ (b) From (a), $T(n) = \Theta(n \log n)$ and $T(n) = \Omega(n \log n)$. So, we have $T(n) = \Theta(n \log n)$.

(For #34 - #42, feel free to use the space based on your need.)