Computer Science
Tompkins High School
December 6th 2014

Directions:

1. DO NOT OPEN THE EXAM UNTIL INSTRUCTED TO DO SO.

2. NO CALCULATORS or calculation devices may be used during the exam.

3. You will have 45 minutes to complete the exam.

4. When time is called you may finish writing down a letter if it is already started.

5. When you are finished with your exam wait quietly.

1. What is the result of the following expression?

$10110_2 + 25_{16}$

A. 74_8 B. 125_{10} C. 111001_2 D. 57_{10} E. $3B_{16}$

2. What is output by the code to the right?

A. 1 B. 5 C. 2

D. 4 E. 8

```
int x = 16;
int y = -8;
int a = x/--y +3;

System.out.println(a);
```

3. What is output by the code to the right?

A. 4 87 B. 4 86 C. 3 87

D. 3 86 E. 3 85

```
int a = 4;
int b = 17;

b += b++ * --a + b;
System.out.println(a + " " + b);
```

4. What integer value(s) can be entered to stop the loop?

A. the value of -1
B. the value of 0
C. values between 5 and 10
D. values greater than 10 or values less than 5
E. No value will stop the loop

```
int a = 0;

do
{
  System.out.print("Enter a value between 5 and 10: ");
  a = kb.nextInt();
}while(a >=5 || a <=10);
```

5. What is output by the code to the right?

A. 5 iso B. 4 iso C. 6 s

D. 6 coo E. 5 oco

```
StringBuffer text = new StringBuffer("Jojo is so cool");

text.replace(10,11,"lo");
text.replace(7,8,"ol");
text.replace(4,5,"lol");

String[] r = text.toString().split("ol");
System.out.print(r.length+" "+r[4]);
```

6. What is output by the code to the right?

A. 34 B. 18 C. 7

D. 8 E. 6

```
int[] a = {6,8,13,5,2};

for(int i=1; i< 5; i++)
  a[i] =  a[i]+a[i-1];
System.out.println(a[4]);
```

7. What is output by the code to the right?

A. A B. AP C. AE

D. PE E. APE

```
int x = 15;
int y =4;

if(++x>15 || y--<8)
  System.out.print("A");
if(y==3 && x>y)
  System.out.print("P");
else
  System.out.print("E");
```

8. What is output by the code to the right?

A. 4 5 true B. 3 5 false C. 3 5 true

D. 4 4 true E. 4 5 false

```
int z = 3;
int x = 2;
int y = 6;
boolean m = (z++ < 5 && x++ > 2) || y-- < 6;
System.out.println(x + " " +y + " " + m);
```

9. What is output by the code to the right?

A. \”\\ // \”” B. “\/” C. \”\\/\”
D. “\//” E. \”\”

```
System.out.print("\"\\//\"”);
```

10. What is output by the code to the right?

A. adgbeh
B. adgbehcfi
C. abcdef
D. abde
E. abcdefghi

```
char[][] test = {        {'a', 'b', 'c'},
                         {'d', 'e', 'f'},
                         {'g', 'h', 'i'}};

for(int c = 0; c<2; c++)
{
   for(int r = 0; r<3;r ++)
      System.out.print(test[r][c]);
}
```

11. Which of the following lines would generate an int value from 17 to 35 (including 17 and 35)?

A. Math.random(17,35);
B. (int)(Math.random()*35-18);
C. (int)(Math.random()*19+17);
D. (int)(Math.random(17,35));
E. (int)(Math.random()*18+17);

12. What is output by the code to the right?

A. #Turtles   #are# ninjas#
B. #   Turtles#are#ninjas #
C. #Turtles#   are #ninjas#
D. #Turtles#are#ninjas#
E. #Turtles   #are#Turtles   #

```
String s = "Turtles";
String t = "are";
String u = "ninjas";
System.out.printf("#%-10s#%s#%8s#",s,t,u);
```

13. What is output by the code to the right?

A. 6   B. 9   C. 7

D. 11   E. 13

```
int[] a = {4,3,6,5,8};

int m = a[0]+a[1];

for(int i=0; i< 3; i++)
   if(m < a[i]+a[i+1])
      m = a[i]+a[i+1];
System.out.print(m);
```

14. What is output by the code to the right?

A. 5   B. 7   C. 2

D. 3   E. 4

```
int[] k = {0,3,2,6,5,3,4,7,8,7,5};
System.out.println(k[k[2]*3]);
```

15. What is output by the code to the right?

A. abaabb   B. abaaab   C. aaaa

D. baabab   E. abaab

```
for(int x = 0; x <= 6; x+=2)
{
   if(x%2==0)
      System.out.print("a");
   if(x%3==0)
      System.out.print("b");
}
```

16. What is output by the code to the right?

A. 0   B. 6   C. 75

D. 9   E. 39

```
int a = 18<<5>>3 + 3;
System.out.println(a);
```

17. What is output by the code to the right?

A. 121212   B. 111333   C. 112233

D. 321321   E. 123123

```
for(int x = 1; x <= 3; x++)
   for(int y=1; y<=2; y++)
      System.out.print(x);
```

18. What is output by the code to the right?

A. 7*true*[8, 3, 14, 19]
B. true*true*[8, 7, 19]
C. true*true*[8, 7, 14, 19]
D. 14*true*[8, 7, 3, 19]
E. Index Out of Bounds Error

```
ArrayList<Integer> numbers = new ArrayList<Integer>();
numbers.add(8);
numbers.add(7);
Integer s = new Integer(14);
numbers.add(s);
numbers.add(s);
numbers.add(3);
numbers.add(19);

System.out.println(numbers.remove(3)+"*"+numbers.remove(s)+"*"+numbers);
```

19. What is output by the code to the right?

A. 12555 B. 52356 C. 52355
D. 125455 E. 523556

```
for(int c=1;c<=6;c++)
  System.out.print((c%3==1)?5:(c<4)?c:++c);
```

20. What code should replace /* 1 */ in the code to the right?

A. (p1.x-p2.x) / (p1.y/p2.y)
B. (p2.getX()-p2.getX()) / (p1.getY()-p2.getY())
C. (p2.y-p1.y) / (p2.x-p1.x)
D. (p2.getY()-p1.getY()) / (p2.getX()-p1.getX())
E. (p2.getY()-p1.getX()) / (p2.getX()-p1.getX())

```
class XPoint
{
  public int x;
  public int y;

  public XPoint(int x, int y)
  {
    this.x=x;
    this.y=y;
  }
}

class XLine
{
  public XPoint p1;
  public XPoint p2;

  public XLine(XPoint p1, XPoint p2)
  {
    this.p1=p1;
    this.p2=p2;
  }

  public int slope()
  {
    return /* 1 */;
  }
}
```

21. What is output of the below code?

```
XPoint a = new XPoint(2,4);
XPoint b = new XPoint(3,3);
XLine c = new XLine(b,a);
System.out.println(c.p2.x + ", "+c.p1.y);
```

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

D. 3, 3 E. 4, 4

22. What is output of the below code?

```
XPoint a = new XPoint(4,7);
XPoint b = new XPoint(5,6);
XLine c = new XLine(a,b);
a.x =18;
XPoint m = new XPoint(5,6);
XLine d = new XLine(a,m);
System.out.print(c.p1.equals(d.p1) + " ");
System.out.print(c.p2.equals(d.p2));
```

A. false false B. false true C. true false

D. true true E. true

23. What is output by the code to the right?

A.193 B. 116 C.75
D. 1F E. 154

```
int x = 0x42 | 0b1110;
System.out.println(Integer.toString(x,8));
```

24. What is output by the code to the right?

A.0 B. 3 C.4
D. 2 E. 5

```
ArrayList<String> text = new ArrayList<String>();
text.add("bo");
text.add("bot");
text.add("box");
text.add("so");
text.add("mpo");
text.add("rock");

for(int x=0; x<text.size(); x++)
 if(text.get(x).length() >=3)
   text.remove(x);
System.out.println(text.size());
```

25. What is output by the code to the right?

A.8 B. 6 C.0
D. 9 E. 5

```
String a = "snakes are slimy";
String b = "snakes are scary";
System.out.println(a.compareTo(b));
```

26. What would the result of a pre-order print after the code the right is run?

A. a, b, r, z, A, C, M
B. A, C, M, a, b, r, z
C. M, A, C, a, r, b, z
D. C, A, b, z, r, a, M
E. A, a, b, C, M, r, z

```
BinarySearchTree<String> bst = new BinarySearchTree<String>();

bst.add(“M”);
bst.add(“a”);
bst.add(“r”);
bst.add(“A”);
bst.add(“b”);
bst.add(“C”);
bst.add(“z”);
```

27. What replaces code should replace /* 1 */ in the code to the right?

A. (index-2)/2
B. (index*2)
C. (index)/2
D. (index-1)/2
E. (index+1)/2

28. What replaces code should replace /* 2 */ in the code to the right?

A. data.set(x,data.set(y,data.get(y)));
B. data.set(x) = data.get(y);
   data.set(y) = data.get(x);
C. data.set(x, data.get(y));
   data.set(y, data.get(x));
D. data.get(x) = data.get(y);
   data.get(y) = data.get(x);
E. data.set(y,data.set(x,data.get(y)));

29. What replaces code should replace /* 3 */ in the code to the right?

A.
data.get(c).compareTo(data.get(getParentIndex(c))) > 0
B.
data.get(c).compareTo(data.get(getParentIndex(c))) < 0
C.
data.get(c).compareTo(data.get(getParentIndex(c))) != 0
D.
data.get(c) < data.get(getParentIndex(c)
E.
data.get(c) < data.get(getParentIndex(c)

30. What is the output of the below code?

```
MaxHeap<Integer> mh = new MaxHeap<Integer>();
mh.add(5);
mh.add(2);
mh.add(12);
mh.add(4);
mh.add(66);
mh.remove();
System.out.println(mh);
```

A. [12, 4, 5, 2]   B. [12, 5, 4, 2]   C. [12, 2, 5, 4]

D. [2, 5, 4, 12]   E. [2, 4, 5, 12]

```
// The following code implements a Maximum Heap
class MaxHeap<E extends Comparable>
{
  private ArrayList <E> data= new ArrayList <E>();

  public boolean add(E item)
  {
    data.add(item);
    heapifyUp();
    return true;
  }

   public E remove()
   {
     if(data.isEmpty())
        return null;
     else
     {
        swap(0,data.size()-1);
        E removed = data.remove(data.size()-1);
        if(!data.isEmpty())
           heapifyDown();
        return removed;
      }
    }

  public String toString()
 {  return data.toString(); }

  public int getParentIndex(int index)
 {   return /* 1*/;  }

 public void swap(int x, int y)
 {
    /* 2 */;
 }

  public void heapifyUp()
  {
    int c = data.size()-1;

    while(c!=0 && /* 3 */)
    {
        swap(c,getParentIndex(c));
        c = getParentIndex(c);
     }
  }

  /* Implementation Not Shown For The Following Methods */
  public int getLeftChild(int index);

  public int getRightChild(int index);

  public void heapifyDown();

}
```

31. What would be the result of recur("go",0)?

A. XgXgXgXoXoXoX
B. goXgo
C. XgXoXoX
D. XgXgXgXoX
E. XoXgX

32. What would be the result of recur("falcons",7)?

A. XsXoXfXlXfXnXfXfXnXcXsX
B. XfXnXaXsX
C. XfXnXaXsXaXsXoXfX
D. aXnXcXfXaXcXoXf
E. XfXnXcXsX

```
public static String recur(String a, int x)
{
  if(x > a.length()*2)
    return "X";
  else
  {
    char g = a.charAt(x%a.length());
    return recur(a,x+2) + g + recur(a,x+3);
  }
}
```

33. What is the out of the code to the right?

A.192 140
B. 70 52
C.524 384
D. 36 84
E. 125 148

```
int a = 0,b = 1,c = 1,d = 4,i = 5;
while(i>0)
{
  a = d;
  b = c;
  c = a+d;
  d = c+b;
  i--;
}
System.out.println(a+" "+b);
```

34. What is the out of the code to the right?

A.74 B. 103 C.74
D. 90 E. The loop never terminates

```
int[] a = {9, 8, 6, 2, 4, 6};
int i = 0;
int x = 3;
int total = 3;
while(i<10)
{
  try
  {
    total += a[i%3]/(a[++x]/a[x++]);
    i++;
  }
  catch  (Exception e)
  {
    total+=3;
    x=x%(x/(x-i%2));
  }
}
System.out.println(total);
```

35. What is the out of the code to the right?

A.3 B. 4 C.5
D. 9 E. 0

```
HashSet<Integer> set = new HashSet<Integer>();
 set.add(new Integer(4));
 set.add(9);
 set.add(8);
 set.add(3);
 set.add(4);
 set.add(9);

System.out.println(set.size());
```

36. What is the out of the code to the right?

A.56 B. 62 C.30
D. -30 E. -62

```
String a = "obligation";
 String b = "19 Dots";
 System.out.println(a.compareTo(b));
```

37. Given a very large list that is already in order what sort would run the fastest after adding 5 new items.

A. Selection B. Heap C. Insertion
D. Quick E. Bubble

38. What is the out of the code to the right?

A.
hammerhammer90
Jojohammer099

B.
hammerhammer90
Jojohammer99

C.
hammerjojo90
Jojohammer099

D.
hammerjojo90
Jojohammer99

E.
hammerhammer9
Jojohammer18

```
public static int dark(Integer y, String a, String b, int z)
{
  a = b;
  b = a;
  y = z+y;
  z =y-z;
  y = y-z;
  System.out.println(a+b+y+z);
  return y-z;
}

public static void main(String[] args)
{
  Integer y=0;
  String a = "Jojo", b="hammer";
  int z = 9;
  System.out.println(a+b+y+z+dark(y,a,b,z));
}
```

39. What is the output of the code below?

M a = new M();

A.QLQ B. LQ C.LQL
D. QL E. Stack Overflow Exception

40. What is the output of the code below?

M a = new N();

A. LLQLLQQLLLLQLLQQLL
B. LQLLLQLQQLL
C. LLQLLQLQLLLQQL
D. Class cast Exception
E. Stack Overflow Exception

```
class M
{
  public M()
  {
    L();
    Q();
  }

  public void L()
  { System.out.print("L"); }

  public void Q()
  {
    System.out.print("Q");
    L();
  }
}

class N extends M
{
  public N()
  {
    L();
    Q();
  }

  public void L()
  { System.out.print("LL"); }

  public void Q()
  {
    super.Q();
    System.out.print("QQ");
    L();
  }
}
```

Standard Classes and Interfaces — Supplemental Reference
(Accessed From: UIL COMPUTER SCIENCE DISTRICT 2 2011)

class java.lang.Object
- o boolean equals(Object other)
- o String toString()
- o int hashCode()

interface java.lang.Comparable<T>
- o int compareTo(T other)
  - Return value < 0 if this is less than other.
  - Return value = 0 if this is equal to other.
  - Return value > 0 if this is greater than other.

class java.lang.Integer implements Comparable<Integer>
- o Integer(int value)
- o int intValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Integer anotherInteger)
- o static int parseInt(String s)

class java.lang.Double implements Comparable<Double>
- o Double(double value)
- o double doubleValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Double anotherDouble)
- o static double parseDouble(String s)

class java.lang.String implements Comparable<String>
- o int compareTo(String anotherString)
- o boolean equals(Object obj)
- o int length()
- o String substring(int begin, int end)
  - Returns the substring starting at index begin and ending at index (end - 1).
- o String substring(int begin)
  - Returns substring(from, length()).
- o int indexOf(String str)
  - Returns the index within this string of the first occurrence of str. Returns -1 if str is not found.
- o int indexOf(String str, int fromIndex)
  - Returns the index within this string of the first occurrence of str, starting the search at the specified index.. Returns -1 if str is not found.
- o charAt(int index)
- o int indexOf(int ch)
- o int indexOf(int ch, int fromIndex)
- o String toLowerCase()
- o String toUpperCase()
- o String[] split(String regex)
- o boolean matches(String regex)

class java.lang.Character
- o static boolean isDigit(char ch)
- o static boolean isLetter(char ch)
- o static boolean isLetterOrDigit(char ch)
- o static boolean isLowerCase(char ch)
- o static boolean isUpperCase(char ch)
- o static char toUpperCase(char ch)
- o static char toLowerCase(char ch)

class java.lang.Math
- o static int abs(int a)
- o static double abs(double a)
- o static double pow(double base, double exponent)
- o static double sqrt(double a)
- o static double ceil(double a)
- o static double floor(double a)
- o static double min(double a, double b)
- o static double max(double a, double b)
- o static int min(int a, in b)
- o static int max(int a, int b)
- o static long round(double a)
- o static double random()
  - Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

interface java.util.List<E>
- o boolean add(E e)
- o int size()
- o Iterator<E> iterator()
- o ListIterator<E> listIterator()
- o E get(int index)
- o E set(int index, E e)
  - Replaces the element at index with the object e.
- o void add(int index, E e)
  - Inserts the object e at position index, sliding elements at position index and higher to the right (adds 1 to their indices) and adjusts size.
- o E remove(int index)
  - Removes element from position index, sliding elements at position (index + 1) and higher to the left (subtracts 1 from their indices) and adjusts size.

class java.util.ArrayList<E> implements List<E>
class java.util.LinkedList<E> implements List<E>, Queue<E>
Methods in addition to the List methods:
- o void addFirst(E e)
- o void addLast(E e)
- o E getFirst()
- o E getLast()
- o E removeFirst()
- o E removeLast()

class java.lang.Exception

class java.util.Stack<E>
- o boolean isEmpty()
- o E peek()
- o E pop()
- o E push(E item)

interface java.util.Queue<E>
- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

class java.util.PriorityQueue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

interface java.util.Set<E>

- o boolean add(E e)
- o boolean contains(Object obj)
- o boolean remove(Object obj)
- o int size()
- o Iterator<E> iterator()
- o boolean addAll(Collection<? extends E> c)
- o boolean removeAll(Collection<?> c)
- o boolean retainAll(Collection<?> c)

class java.util.HashSet<E> implements Set<E>

class java.util.TreeSet<E> implements Set<E>
interface java.util.Map<K,V>

- o Object put(K key, V value)
- o V get(Object key)
- o boolean containsKey(Object key)
- o int size()
- o Set<K> keySet()
- o Set<Map.Entry<K, V>> entrySet()

class java.util.HashMap<K,V> implements Map<K,V>

class java.util.TreeMap<K,V> implements Map<K,V>

interface java.util.Map.Entry<K,V>

- o K getKey()
- o V getValue()
- o V setValue(V value)

interface java.util.Iterator<E>

- o boolean hasNext()
- o E next()
- o void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>
Methods in addition to the Iterator methods:

- o void add(E e)
- o void set(E e)