UIL COMPUTER SCIENCE WRITTEN TEST

2017 STATE

APRIL 2017

General Directions (Please read carefully!)

1. DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO.
2. There are 40 questions on this contest exam. You will have 45 minutes to complete this contest.
3. All answers must be legibly written on the answer sheet provided. Indicate your answers in the appropriate blanks provided on the answer sheet. Clean erasures are necessary for accurate grading.
4. You may write on the test packet or any additional scratch paper provided by the contest director, but NOT on the answer sheet, which is reserved for answers only.
5. All questions have ONE and only ONE correct answer. There is a 2-point penalty for all incorrect answers.
6. Tests may not be turned in until 45 minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your test until told to do otherwise. You may use this time to check your answers.
7. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
8. All provided code segments are intended to be syntactically correct, unless otherwise stated. You may also assume that any undefined variables are defined as used.
9. A reference to many commonly used Java classes is provided with the test, and you may use this reference sheet during the contest. AFTER THE CONTEST BEGINS, you may detach the reference sheet from the test booklet if you wish.
10. Assume that any necessary import statements for standard Java SE packages and classes (e.g., `java.util`, `System`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.
11. NO CALCULATORS of any kind may be used during this contest.

Scoring

1. Correct answers will receive **6 points**.
2. Incorrect answers will lose **2 points**.
3. Unanswered questions will neither receive nor lose any points.
4. In the event of a tie, the student with the highest percentage of attempted questions correct shall win the tie.

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

package java.lang

```
class Object
  boolean equals(Object anotherObject)
  String toString()
  int hashCode()

interface Comparable<T>
  int compareTo(T anotherObject)
    Returns a value < 0 if this is less than anotherObject.
    Returns a value = 0 if this is equal to anotherObject.
    Returns a value > 0 if this is greater than anotherObject.

class Integer implements Comparable<Integer>
  Integer(int value)
  int intValue()
  boolean equals(Object anotherObject)
  String toString()
  String toString(int i, int radix)
  int compareTo(Integer anotherInteger)
  static int parseInt(String s)

class Double implements Comparable<Double>
  Double(double value)
  double doubleValue()
  boolean equals(Object anotherObject)
  String toString()
  int compareTo(Double anotherDouble)
  static double parseDouble(String s)

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

class Character
  static boolean isDigit(char ch)
  static boolean isLetter(char ch)
  static boolean isLetterOrDigit(char ch)
  static boolean isLowerCase(char ch)
  static boolean isUpperCase(char ch)
  static char toUpperCase(char ch)
  static char toLowerCase(char ch)

class Math
  static int abs(int a)
  static double abs(double a)
  static double pow(double base, double exponent)
  static double sqrt(double a)
  static double ceil(double a)
  static double floor(double a)
  static double min(double a, double b)
  static double max(double a, double b)
  static int min(int a, int b)
  static int max(int a, int b)
  static long round(double a)
  static double random()
    Returns a double greater than or equal to 0.0 and less than 1.0.
```

package java.util

```
interface List<E>
class ArrayList<E> implements List<E>
  boolean add(E item)
  int size()
  Iterator<E> iterator()
  ListIterator<E> listIterator()
  E get(int index)
  E set(int index, E item)
  void add(int index, E item)
  E remove(int index)

class LinkedList<E> implements List<E>, Queue<E>
  void addFirst(E item)
  void addLast(E item)
  E getFirst()
  E getLast()
  E removeFirst()
  E removeLast()

class Stack<E>
  boolean isEmpty()
  E peek()
  E pop()
  E push(E item)

interface Queue<E>
class PriorityQueue<E>
  boolean add(E item)
  boolean isEmpty()
  E peek()
  E remove()

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
  boolean add(E item)
  boolean contains(Object item)
  boolean remove(Object item)
  int size()
  Iterator<E> iterator()
  boolean addAll(Collection<? extends E> c)
  boolean removeAll(Collection<?> c)
  boolean retainAll(Collection<?> c)

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements Map<K,V>
  Object put(K key, V value)
  V get(Object key)
  boolean containsKey(Object key)
  int size()
  Set<K> keySet()
  Set<Map.Entry<K, V>> entrySet()

interface Iterator<E>
  boolean hasNext()
  E next()
  void remove()

interface ListIterator<E> extends Iterator<E>
  void add(E item)
  void set(E item)

class Scanner
  Scanner(InputStream source)
  Scanner(String str)
  boolean hasNext()
  boolean hasNextInt()
  boolean hasNextDouble()
  String next()
  int nextInt()
  double nextDouble()
  String nextLine()
  Scanner useDelimiter(String regex)
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2017 STATE

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 8 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the System class has been statically imported using:**

```
import static java.lang.System.*;
```

Question 1.

Which of the following is the product of 1110_2 and 11101_2?

A) 307_{10} **B)** 12110_4 **C)** $1A6_{16}$ **D)** 1120_7 **E)** 110010111_2

Question 2.

Which of the following is the output of the line of code shown on the right?

A) -2 **B)** 2 **C)** 0 **D)** 4 **E)** -6

```
out.println(36%-8+-14%3);
```

Question 3.

Which of the following is the output of the code shown on the right? *Asterisk * indicate blank spaces.*

A) `Current value is -0000001858`

B) `Current value is -000001858`

C) `Current value is 0000001858`

D) `Current value is *****-1858`

E) `Current value is -1858*****`

```
out.printf("Current value is %0+10d",-1858);
```

Question 4.

Which of the following is the output of this code segment?

A) `OutOfBoundsExceptio`

B) `oundsExcepti`

C) `OutOfBoundsExcepti`

D) `OutO`

E) `oundsExceptio`

```
String
s="StringIndexOutOfBoundsException";
out.print(s.substring(s.indexOf("o"),
          s.lastIndexOf("o")));
```

Question 5.

What is printed by the code segment shown on the right?

A) `true`

B) `false`

```
int x=10,y=12,z=15;
boolean b=x==z||x>=y^z>y;
out.print(b);
```

Question 6.

What is printed by the line of code shown on the right?

A) `3.0` **B)** `3` **C)** `4.0` **D)** `4` **E)** `5.0`

```
out.print(Math.log10(10000));
```

Question 7.

What is the output of the code segment shown on the right?

A) `22 0b1101`

B) `22 13`

C) Error. Throws a run time exception due to an out of range error.

D) Error. Will not compile because of a type mismatch.

E) Error. Invalid number system designation.

```
short s=9;
int i=0b11+0xA;
s=s+i;
out.print(s+" "+i);
```

Question 8.

What is the output of the code segment shown on the right?

A) `BIG DATA`

B) `Big Data`

C) `big data`

D) `a`

E) Error. StringIndexOutOfBoundsException

```
String s="Big Data";
if(s.length()<=7)
   if(Character.isUpperCase(s.charAt(0)))
         out.print(s.toUpperCase());
   else
         out.print(s.toLowerCase());
else
   if(Character.isAlphabetic(s.charAt(0)))
         out.print(s);
   else
         out.print(s.charAt(s.length()-1));
```

Question 9.

What is the output of the code segment shown on the right?

A) Error. Will not compile due to the incomplete `for` statement

B) `# # # # # # 8`

C) `# # # # # 9`

D) `# # # # # # 9`

E) Error. Will not compile because final print statement is not within the scope of varialble f.

```
int f=3;
for(;f<9;){
     out.print("# ");
     f++;
}
out.print(f);
```

Question 10.

What is the output of the code segment shown on the right?

A) Error. Throws an ArrayIndexOutOfBoundsException

B) `0 1 2 3 4 5`

C) `3 0 1 4 2 5`

D) `1 2 4 0 3 5`

E) `5 3 0 4 2 1`

```
int a[]={3,0,1,4,2,5};
int[] b=new int[6];
for(int i=0;i<a.length;i++)
     b[a[i]]=i;
for(int i:b)
     out.print(i+" ");
```

Question 11.

What is the output of the class shown if `datafile.dat` contains the following sentence?

The total time allowed is two minutes.

```
import static java.lang.System.*;
import java.io.*;
import java.util.*;
public class Question11 {

public static void main(String[] args) throws IOException{
      Scanner f=new Scanner(new File("datafile.dat"));
      f.useDelimiter("t");
      while(f.hasNext())
            out.print(f.next());
      }
}
```

A) `The total time allowed is two minutes.`

B) `he oal ime allowed is wo minues.`

C) `The oal ime allowed is wo minues.`

D) `The  o al  ime allowed is  wo minu es.`

E) `Theoalimeallowediswominues.`

Question 12.

What is the output of the code segment shown on the right?

A) `4 3 7`

B) `4 4 8`

C) `4 4 9`

D) `4 4 0`

E) Error. Will not compile.

```
boolean
b[]={true,false,true,true,false,
true,false,false};
int x=0,y=0,z=0;
while(z<b.length){
     if(b[z])x++;
     else y++;
     z++;
}
out.print(x+" "+y+" "+z);
```

Question 13.

What is the output of the line of code shown on the right?

A) `-33`

B) `-34`

C) `-5`

D) `32`

E) `-4`

```
out.print(~18+(int)14.95);
```

Question 14.

What is the output of the code segment shown on the right?

A) `125 126 127`

B) `125 126 127 128`

C) `125 126 127 -128`

D) Unknown because the code segment creates an infinite loop.

E) No output. Throws an exception when variable b goes beyond the range of the byte data type.

```
byte b;
for(b=125;b>0;b++)
   out.print(b+" ");
```

Question 15.

What is the output of the code segment shown on the right?

A) `20 4 true`

B) `30 4 false`

C) `30 3 true`

D) `20 3 false`

E) `30 3 false`

```
ArrayList<Integer> a=new ArrayList<Integer>();
a.add(10);a.add(20);a.add(30);
a.add(40);a.add(50);
out.print(a.get(2)+" ");
a.remove(3);
out.print(a.indexOf(50)+" ");
out.print(a.contains(40));
```

Question 16.

Which of the following statements is false?

A) An identifier is a sequence of characters that consists of letters, digits, underscores (_)and dollar signs ($).

B) An identifier cannot start with a digit.

C) An identifier cannot be a reserved word.

D) An identifier does not have a maximum length.

E) None of the above statements are false.

Question 17.

What is the output of the code segment shown on the right?

A) `f`

B) `i`

C) `a`

D) `c`

E) `d`

```
String let="Honorificabilitudinitatibus";
int n=0;
char b[][][]=new char[3][3][3];
for(int i=0;i<3;i++)
      for(int j=0;j<3;j++)
            for(int k=0;k<3;k++){
                  b[i][j][k]=let.charAt(n);
                  n++;}
out.println(b[0][2][2]);
```

Question 18.

What is the output of the code segment shown on the right?

A) 0

B) 3

C) 12

D) 4

E) 2

Question 19.

If the following lines of code are added to the end of the code segment shown on the right, what would be the additional output?

```
Object o=list.get(1);
out.println(o);
```

A) [a]

B) [d, c, b, a]

C) [a, b, c, d]

D) The hexadecimal value representing the memory location of object o.

E) There is no output due to an error.

Question 20.

If the following lines of code are added to the end of the code segment shown on the right, what would be the additional output?

```
Queue<Double> ll=list.get(2);
out.println(ll.remove());
```

A) [1.61, 1.41, 2.71, 3.14]

B) [3.14, 2.71, 1.41, 1.61]

C) 1.61

D) 3.14

E) There is no output due to an error.

```
List<Object> list=new LinkedList<Object>();
ArrayList<Integer> al=new
ArrayList<Integer>();
al.add(1);al.add(2);al.add(3);al.add(4);
Stack<String> s=new Stack<String>();
s.push("a");s.push("b");s.push("c");
s.push("d");
Queue<Double> li=new LinkedList<Double>();
li.add(3.14);li.add(2.71);li.add(1.41);
li.add(1.61);
list.add(al);list.add(s);list.add(li);
out.println(list.size());
```

Question 21.

Which of the following values for string variable s will make this line of code print false?

```
out.print(s.matches("[a-z&&[^aeiou]]+\\d?"));
```

A) bc43

B) bc4

C) bc

D) jklmn9

E) None of the above will make the line of code print false.

Question 22.

What is printed by the line of code shown on the right?

A) 32 **B)** 1A **C)** 20 **D)** 10100 **E)** 26

```
out.print(011+0x11);
```

Question 23.

What is printed by **line #1** in the client code shown on the right?

A) C

B) D

C) CD

D) DC

E) CDCD

Question 24.

What is the output of **line #2** in the client code shown on the right?

A) C

B) D

C) CD

D) DC

E) CDCD

```
public class A {
      private String c="C";
      public String d(){return c;}
}
public class B extends A {
      private String c="D";
      public String d() {return c+e();}
      public String e() {return super.d();}
}
//client code
A a=new A();
out.println(a.d());//line #1
B b=new B();
out.println(b.d());//line #2
```

Question 25.

Consider the hash map `lhm` declared in the line of code shown on the right. Which of the following segments of code will **not** print all of the entries in that map? Disregard the format of the output.

A)
```
Set s=lhm.keySet();
for(Object keys:s)
out.println(keys+" "+lhm.get(keys));
```

B)
```
Set s2=lhm.entrySet();
out.println(s2);
```

C)
```
out.println(lhm);
```

D)
```
Set s=lhm.keySet();
for(String keys:s)
out.println(keys+" "+lhm.get(keys));
```

E) All of the above will correctly print all of the entries in the map `lhm`

```
LinkedHashMap<String,Integer> lhm=new
LinkedHashMap<String,Integer>();
```

Question 26.

Which of the following values will not be printed by the code segment shown on the right?

A) -6

B) 4

C) 2

D) 0

E) -2

```
double ran=Math.random();
int i=(int)(ran*5-3)*2;
out.print(i);
```

Question 27.

What is the output of this call to method `abc` shown on the right?

```
out.print(abc(4));
```

A) 15

B) 6

C) 12

D) 1

E) 24

```
public static int abc(int x){
      if(x<0)
            return 0;
      else if(x==0)
            return 1;
      else
            return 1+abc(x-1)+abc(x-2);
}
```

Question 28.

Given the method `abc` shown on the right, what is the output of the call to `abc` shown here?

```
out.print(Arrays.toString(xyz(2,3,4)));
```

A) [0, 0, 3, 5, 2, 2, 2, 2]

B) [0, 0, 5, 5, 2, 2, 2, 2]

C) [0, 0, 0, 2, 2, 2, 2, 2]

D) [5, 7, 2, 2, 2, 2]

E) [5, 5, 2, 2, 2, 2]

```
public static int[] xyz(int x, int y, int z){
     int[] a=new int[8];
     while(z<a.length){
       a[z]=x*y/3;
       a[z/2]=x+y&z;
       z++;
     }
     return a;
   }
```

//Use the following code to answer questions 29, 30, 31 and 32.

```
public static void quickSort(int[] list, int first, int last){
      if(last>first){
            int pivotIndex=partition(list,first,last);
            quickSort(list,first,pivotIndex-1);
            <code 3>
      }
}
public static int partition(int[] list, int first, int last){
      int pivot=list[first];
      int low=first+1;
      int high=last;
      while(high>low)
            {
            while(<code 1>) low++;
            while(<code 2>) high--;
            if(high>low)
                  {
                  int temp=list[high];
                  list[high]=list[low];
                  list[low]=temp;
                  }
            }
      while(high>first&&list[high]>=pivot) high--;
      if(pivot>list[high])
            {
            list[first]=list[high];
            list[high]=pivot;
            return high;
            }
      else
            return first;
      }
```

Question 29.

The methods listed above is intended to implement the Quicksort algorithm. What must replace **<code 1>** and **<code 2>** to ensure that the `partition` method correctly sorts `list` in ascending order?

A) `low<=high&&list[low]>=pivot`
`low<=high&&list[high]<pivot`

B) `low<=high||list[low]<=pivot`
`low<=high||list[high]>pivot`

C) `low<=high`
`low<=high`

D) `list[low]<=pivot`
`list[high]>pivot`

E) `low<=high&&list[low]<=pivot`
`low<=high&&list[high]>pivot`

Question 30.

Assume that **<code 1>** and **<code 2>** have been filled in correctly. What must replace **<code 3>** to ensure that the `quickSort` method correctly sorts `list` in ascending order?

A) `quickSort(list,pivotIndex+1,last);`

B) `quickSort(list,last,pivotIndex+1);`

C) `quickSort(list,pivotIndex,last);`

D) `pivotIndex=partition(list,last,first);`

E) No additional code is required at this point.

Question 31.

Which of the following is the base case condition for the method `quickSort`?

A) `first` equals `last`

B) `last` is greater than `first`

C) `pivotIndex` is equal to zero

D) `pivotIndex` is equal to last

E) `high` is equal to `low`

Question 32.

What is the best, average and worst case time complexity (Big O value) for the Quicksort algorithm?

A) $O(n^2)$ $O(n^2)$ $O(n^2)$

B) $O(\log n)$ $O(\log n)$ $O(n^2)$

C) $O(n \log n)$ $O(n \log n)$ $O(n^2)$

D) $O(n \log n)$ $O(n \log n)$ $O(n \log n)$

E) $O(n)$ $O(n \log n)$ $O(n^2)$

```
//Use the following code to answer questions 33, 34 and 35

public class HeapNode {
      private int data;
      public HeapNode(int n){data=n;}
      public int getData(){return data;}
}

public class Heap {
      private HeapNode[] heap;
      private int max;
      private int cur;

      public Heap(int size){
            max=size;
            cur=0;
            heap=new HeapNode[max];}

      public void moveUp(int index){
            //missing implementation
      }

      public void moveDown(int index){
            int largerChild;
            HeapNode top=heap[index];
            while(index<cur/2){
                  int left=2*index+1;
                  int right=left+1;
                  if(right<cur&&heap[left].getData()<heap[right].getData())
                        largerChild=right;
                  else
                        largerChild=left;
                  if(<code 1>)
                        break;
                  heap[index]=heap[largerChild];
                  index=largerChild;}
            heap[index]=top;}

      public boolean isEmpty(){return cur==0;}

      public boolean insert(HeapNode hn){
            if(cur==max) return false;
            heap[cur]=hn;
            moveUp(cur++);
            return true;}

      public HeapNode remove(){
            HeapNode root=heap[0];
            heap[0]=heap[--cur];
            moveDown(0);
            return root;}

      public String toString(){
            String temp="";
            for(int i=0;i<cur;i++)
                  temp+=heap[i].getData()+" ";
            return temp;}
}
```

Question 33.

The classes Heap and HeapNode are a partial implementation of a max heap data structure. Which of the following code segments is the correct implementation of the `moveUp` method?

A.
```
int parent=(index-1)/2;
HeapNode bottom=heap[index];
while(index<0||heap[parent].get
Data()>bottom.getData()){
   heap[index]=heap[parent];
   index=parent;
   parent=(parent-1)/2;}
heap[index]=bottom;
```

B.
```
int parent=2*index+1;
HeapNode bottom=heap[index];
while(index>0&&heap[parent].get
Data()<bottom.getData()){
   heap[index]=heap[parent];
   index=parent;
   parent=2*parent+1;}
heap[index]=bottom;
```

C.
```
int parent=(index-1)/2;
HeapNode bottom=heap[index];
while(index>0&&heap[parent].get
Data()<bottom.getData()){
   heap[parent]=heap[index];
   parent=index ;
   index=(parent-1)/2;}
heap[index]=bottom;
```

D.
```
int parent=(index-1)/2;
HeapNode bottom=heap[index];
while(index>0&&heap[parent].get
Data()<bottom.getData()){
   heap[index]=heap[parent];
   index=parent;
   parent=(parent-1)/2;}
heap[index]=bottom;
```

E.
```
int parent=(index-1)/2;
HeapNode bottom=heap[index];
while(index>0&&heap[parent].get
Data()<bottom.getData()){
   heap[index]=heap[parent];
   index=parent;
   parent=(parent-1)/2;}
bottom=heap[index];
```

Question 34.

Which of the following must replace **<code 1>** in the `moveDown` method so that it will compile and execute correctly within the implementation of a max heap?

A) `top.getData()==heap[largerChild].getData()`

B) `top.getData()>=heap[largerChild].getData()`

C) `top.getData()>=largerChild`

D) `top==largerChild`

E) `top.getData()<heap[largerChild].getData()`

Question 35.

Assume that the `moveUp` method has been properly implemented and that **<code 1>** has been filled in correctly. What is the output of the client code shown on the right?

A) `0 1 3 4 7 10`

B) `10 7 4 3 1 0`

C) `1 7 10 0 3 4`

D) `0 10 1 7 3 4`

E) `10 3 7 0 1 4`

```
Heap h=new Heap(6);
h.insert(new HeapNode(1));
h.insert(new HeapNode(7));
h.insert(new HeapNode(10));
h.insert(new HeapNode(0));
h.insert(new HeapNode(3));
h.insert(new HeapNode(4));
System.out.println(h);
```

Question 36.

Which of the following is equivalent to $A * \bar{B} + \bar{A} * B$?

A) $\overline{A \oplus B}$ **B)** $A \oplus B$ **C)** $A * B$ **D)** $\overline{A * B}$ **E)** $A + B$

Question 37.

What is the value of the postfix expression shown here? (The operands are 1, 2, 9 and 5.)

1 2 + 9 * 5 –

A) 22 **B)** 14 **C)** 48 **D)** 105 **E)** – 42

Question 38.

The method shown on the right will hash a string to a key value to be used in a hash table. If `arraySize` is 100, what is the largest possible value that could be returned by `hashFun`?

A) 312

B) 100

C) 99

D) 27^{100}

E) 100^{27}

```
public static long hashFun(String key){
long hashVal=0;
for(int j=0;j<key.length();j++){
long letter=
   Character.toLowerCase(key.charAt(j))-96;
hashVal=(hashVal*27+letter)%arraySize;
}
return hashVal;
}
```

Question 39.

Find the sum of 11001101 and 11100011. Both values are shown as signed 8-bit two's complement binary numbers. Write your answer as a decimal number.

Question 40.

How many edges does a complete graph with 12 nodes contain?

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2017 STATE

Questions (+6 points for each correct answer, -2 points for each incorrect answer)

1) D	11) C	21) A	31) A
2) B	12) B	22) E	32) C
3) B	13) C	23) A	33) D
4) B	14) A	24) D	34) B
5) A	15) E	25) D	35) E
6) C	16) E	26) B	36) B
7) D	17) D	27) C	37) A
8) B	18) B	28) B	38) C
9) D	19) C	29) E	*39) -80
10) D	20) E	30) A	*40) 66

** See "Explanation" section below for alternate, acceptable answers.*

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 8 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

This page intentionally left blank

Explanations:

1. D $1110_2 = 14_{10}$, $11101_2 = 29_{10}$, 14 X 29 = 406_{10}, $12110_4 = 404_{10}$,$1A6_{16} = 422_{10}$, $1120_7 = 406_{10}$, $110010111_2 = 407_{10}$
2. B 36 % -8 + -14 % 3 = 4 + (-2) = 2. Modulus with negative numbers always takes the sign of the dividend.
3. B The 0 (zero) flag places leading zeroes in the spaces not used by a number. The + flag forces the inclusion of a + or – sign for numerical output.
4. B s.indexOf(o) returns 17. s.lastIndexOf(o) returns 29. s.substring(17,29) returns a substring from character 17 to character 28.
5. A 10 == 15 || 10 >= 12 ^ 15 > 12, False || False ^ True, False || True, True
6. C Math.log10(x) returns the power of 10 equal to x as a double. $10^4 = 10000$.
7. D 0b11 is 3 decimal and 0xA is 16 decimal. 3 + 16 = 19. 19 will certainly fit within the short data type however an int type variable might store a value that exceeds the range of the short data type. Therefore there is a type mismatch error.
8. B The length of s is 8. Character.isAlphabetic(c) returns true if the character is a Unicode alphabet character, false otherwise.
9. D A for statement does not require initialization or incrementation within the statement itself. f is declared outside the scope of the for loop so it can be used in the print statement. A # is printed for each of the values 3, 4, 5, 6, 7, and 8 then f is incremented to 9 and the loop terminates.
10. D

b[a[0]]	b[3] = 0
b[a[1]]	b[0] = 1
b[a[2]]	b[1] = 2
b[a[3]]	b[4] = 3
b[a[4]]	b[2] = 4
b[a[5]]	b[5] = 5

11. C f.next returns strings between each occurrence of a lower case t, not including the t.
12. B x is counting the true values, y is counting the false values and z is the total of all values in array b.
13. C ~ is the complement operator. Add one and take the opposite and you get -19. Casting to int results in a value of 14. -19+14 = -5.
14. A The range of possible values for the byte data type is -128 to 127. When b is incremented beyond 127 the value “wraps around” and becomes -128. -128 is less than 0 so the loops stops.
15. E a.get(2) returns but does not remove 30. a.remove(3) returns and removes 40 but does not print it. After removing 40, a.indexOf(50) returns 3. 40 is no longer in the array list so s.contains(40) returns false.
16. E Answer choices A through D are all true statements about Java identifiers.
17. D b is a 3 X 3 cube. b is filled by placing characters by cell front to back (k), then by column left to right (j), and finally by row top to bottom (i). b[0][2][2] is in the first row, third column, third cell back.
18. B Although each of the data structures added to list contains several elements, each is just one element within list.
19. C The second element in list is the stack s. The Stack class contains a toString method that allows it to be printed where the top of the stack is printed last.
20. E list.get(2) returns an Object which must first be cast as follows: (Queue<Double>). There is a type mismatch.
21. A For a string to match the regular expression shown it must begin with one or more lower case letters except vowels ([a-z&&[^aeiou]]+) and be followed by zero or 1 digit (\\d?). Answer choice A ends with 2 digits.
22. E A leading zero indicates an octal number. $011_8 = 9_{10}$. 0x indicates a hexadecimal number. $0x11_{16} = 17_{10}$. 9 + 17 = 26. The print method will print a decimal number regardless of the number system used in the expression.
23. A Object a is of type A so it calls the method d from the class A which returns the value stored in field c.
24. D Field c and method d within the class B override the field and method with the same name that are inherited from class A. So method d in the class B returns the value stored in class B’s field c, which is “D”. The call to e in class B subsequently calls method d from class A which returns a “C”.
25. D Answer choice D creates a type mismatch. Elements within set s are of type Object.
26. B Math.random() returns a double value x such that 0.0<=x<1.0. To print 4 ran would have to have a value of 1.0.

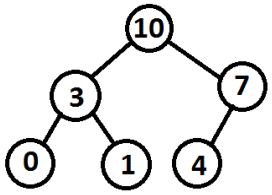
27. C The method adds one for each value of x except when x is less than zero. Here is the call stack.

```
x=4
x=3
x=2
x=1
x=0
x=-1
x=0
x=1
x=0
x=-1
x=2
x=1
x=0
x=-1
x=0
```

28. B

x=2 y=3 z=4 [0, 0, 4, 0, 2, 0, 0, 0]
x=2 y=3 z=5 [0, 0, 5, 0, 2, 2, 0, 0]
x=2 y=3 z=6 [0, 0, 5, 4, 2, 2, 2, 0]
x=2 y=3 z=7 [0, 0, 5, 5, 2, 2, 2, 2]

29. E The first while loop iterates from the front of the list until it finds a value greater than the pivot value. The second while loop iterates from the back of the list until it finds a value less than the pivot value.
30. A The salient feature of a Quicksort is two succesive recursive calls. The first with the front partition and the second with the back partition.
31. A The the base case is the condition that when true stops the execution of the method. When first equals last, last is no longer greater than first, list is completely sorted and the method terminates.
32. C
33. D (index-1)/2 will calculate the index value of a child nodes parent within the backing array. While not at the top of the heap and the child node is larger than its parent move the child up.
34. B If the top node is larger than the largest of its children, it is in its proper place and the method stop execution.
35. E A max heap is a binary tree data structure that ensures that every node is larger than each of its children and that the tree is complete. The implementation shown on the test uses an array as the backing structure. The root node is always at index 0. To calculate the index of each child use 2*index+1 and 2*index+2. Here is a diagram of the heap shown as a tree.



36. B Identity for Exclusive OR
37. A Same as (1 + 2) * 9 – 5
38. C Each recalculation of the hash value within the for loop includes modulus division by 100. Therefore the value returned by hashFun can never be greater than 99 regardless of the value of key.
39. -80 Both values are negative because their sign bits are 1. To convert to decimal take the complement, add one and convert to decimal. (-51)+(-29)=-80
40. 66 The formula to calculate edges is 1/2n(n-1). ½*12*11=66.