**University of Texas at Austin**

**High School Computer Science Competition - 2013**

General Directions:

1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.

2) **No calculator of any kind may be used.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card which are reserved for answers only.

7) You may use additional scratch paper attached to the back of this exam.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers. **All provided code segments are intended to be syntactically correct, unless syntax error is an option among the answers.**

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets and scratch paper from the test booklet, but do not do so until the contest begins.

10) Assume that any necessary import for standard Java packages and classes (e.g. .util, ArrayList, etc.) are included in any programs or code segments that refer to methods from these classes and packages.

Scoring:

1) All questions will receive **6 points** if answered correctly; no points will be given or subtracted if unanswered; **2 points** will be deducted for an incorrect answer.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question 1** | | | | |  | | | | | | | E | | | | | | | | | | | |
|  | What does 111111102 minus AB16 equal? | | | | | | | | | | | | | | | | | | | | | | |
|  | A. | | 1A916 | | | | B. | | | | F116 | | | | | C. | AB16 | | D. | 6516 | E. | 5316 | |
| **Question 2** | | | | | |  | | | | | | | | B | | | | | int x2;  x2 = 5 \* 4 + 37 / 10 - 2 \* 2 \* 2 \* 2 \* 2; System.out.print(x2); | | | | |
|  | | | What is output by the code to the right? | | | | | | | | | | | | | | | |
|  | | | A. | | 336 | | B. | | | -2 | | | | | C. | -5 | | |
|  | | | D. | | -9 | | E. | | | -27 | | | | | | | | |
| **Question 3** | | | | | |  | | | | | | | | B | | | | | int x3 = 0;  for(int j = 0; j <= 3; j++) {  x3++;  ++x3;  }  x3++;  System.out.print(x3); | | | | |
|  | | | What is output by the code to the right? | | | | | | | | | | | | | | | |
|  | | | A. | | 3 | | | | B. | | 6 | | | | | | | |
|  | | | C. | | 7 | | | | D. | | 9 | | | | | | | |
|  | | | E. | | There is no output due to a syntax error. | | | | | | | | | | | | | |
| **Question 4** | | | | | |  | | | | | | | | B | | | | | String st1 = "The\_Tower";  String st2 = st1.substring(6, 9);  String st3 = st1 + st2;  System.out.print(st3.length()); | | | | |
|  | | | What is output by the code to the right? | | | | | | | | | | | | | | | |
|  | | | A. | | 3 | | B. | | | 6 | | | | | C. | 8 | | |
|  | | | D. | | 11 | | E. | | | 12 | | | | | | | | |
| **Question 5** | | | | | |  | | | | | | | | B | | | | | double[] ds = {.1, 5, 0.0, 0, 1.5, 2.1, 0};  System.out.print(ds.length);  System.out.print(" " + (ds[1] + ds[3])); | | | | |
|  | | | What is output by the code to the right? | | | | | | | | | | | | | | | |
|  | | | A. | | 7 .1 | | B. | | | 6 0.1 | | | | | C. | 6 5 | | |
|  | | | D. | | 7 5.0 | | E. | | | 7 0.1 | | | | | | | | |
| **Question 6** | | | | | |  | | | | | | | | B | | | | | int x6 = 53;  int y6 = x6 % 17 + 17 % x6;  System.out.print(y6); | | | | |
|  | | | What is output by the code to the right? | | | | | | | | | | | | | | | |
|  | | | A. | | 57 | | B. | | | 56 | | | | | C. | 55 | | |
|  | | | D. | | 20 | | E. | | | 19 | | | | | | | | |
| **Question 7** | | | | | |  | | | | | | | | B | | | | | boolean p, q, r;  // code in initialize p, q, and r  boolean s = !(p && !q) || !r; | | | | |
|  | | | How many combinations of values of the boolean variables p, q, and r will result in s being set to false? | | | | | | | | | | | | | | | |
|  | | | A. | | 1 | | B. | | | 3 | | | | | C. | 4 | | |
|  | | | D. | | 7 | | E. | | | 8 | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question 8** | | |  | | | B | | | System.out.print(0);  int x = 4;  int y = 8;  if(y < x)  System.out.print(1);  if(x / y == 0)  System.out.print(2);  else if(x != y)  System.out.print(3); |
|  | What is output by the code to the right? | | | | | | | |
|  | A. | 01 | | B. | 013 | | C. | 1 |
|  | D. | 12 | | E. | 02 | | | |
| **Question 9** | | |  | | | D | | | public class Residence {  private int id;  public Residence() { id = -1; }  public Residence(int i) { id = i; }  public String toString() {  return "" + id;  }  }  public class Dorm extends Residence {  private int rooms;  public Dorm() { rooms = -2; }  public Dorm(int r) {  super(r \* 2);  rooms = r;  }  public String toString() {  return rooms + super.toString();  }  } |
|  | Given classes Residence and Dorm to the right, what is output by the following client code?  Residence r1 = new Residence(); Residence r2 = new Residence(-2); System.out.print(r1 + " " + r2); | | | | | | | |
|  | A. | 0 -2 | | | | | | |
|  | B. | id id | | | | | | |
|  | C. | -1 -2 | | | | | | |
|  | D. | -1 2 | | | | | | |
|  | E. | There is no output due to a syntax error in the code. | | | | | | |
| **Question 10** | | |  | | | D | | |
|  | Given classes Residence and Dorm to the right, what is output by the following client code?  Residence r3 = new Dorm(); Dorm d4 = new Dorm(150); System.out.print(r3 + " " + d4); | | | | | | | |
|  | A. | -1 150300 | | | | | | |
|  | B. | -1 300150 | | | | | | |
|  | C. | -20 300150 | | | | | | |
|  | D. | -2-1 150300 | | | | | | |
|  | E. | There is no output due to a syntax error in the code. | | | | | | |
| **Question 11** | | |  | | | B | | | int m = 72;  int n = 63;  System.out.print(n & m); |
|  | What is output by the code to the right? | | | | | | | |
|  | A. | 72 | | B. | 64 | | C. | 63 |
|  | D. | 8 | | E. | 0 | | | |
| **Question 12** | | |  | | | B | | | double x12 = 3.7;  double y12 = Math.floor(x12) +  Math.ceil(x12);  System.out.print(y12); |
|  | What is output by the code to the right? | | | | | | | |
|  | A. | 0.0 | | B. | 3.7 | | C. | 6.0 |
|  | D. | 6.4 | | E. | 7.0 | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question 13** | | |  | | | | | B | | | System.out.print("A");  System.out.println("CB"); |
|  | What is output by the code to the right? | | | | | | | | | |
|  | A. | ACB | | B. | | A C B | | | C. | A CB |
|  | D. | A  CB | | E. | | AC B | | | | |
| **Question 14** | | |  | | | | | B | | | double x14 = -12.50;  System.out.printf("val:%07.1f", x14); |
|  | What is output by the code to the right? | | | | | | | | | |
|  | A. | -0012.5 | | | B. | | val: -12.5 | | | |
|  | C. | val:-12.500 | | | D. | | -12.50 | | | |
|  | E. | val:-0012.5 | | | | | | | | |
| **Question 15** | | |  | | | | | B | | | public int m1(int x, int y) {  x \*= 2;  y /= 2;  return x + y;  }  public int m2(int x) {  int y = x % 10;  int z = m1(y, x);  return x + z - y;  }  public int m3(int x, int y) {  int z = m1(m2(x), m2(y));  z -= 10;  return z - 10;  }  // client code  int x = 2;  int y = 3;  int z = m1(x, y);  System.out.print(x + " " + y + " " + z);//1  x = 12;  y = m2(x);  System.out.print(x + " " + y); // 2  x = 0;  y = 0;  System.out.print(m3(x, y)); // 3  z = 5;  x = 3;  y = m3(x, z);  System.out.print(x+ " " + y + " " + z); //4 |
|  | What is output by the line in the client code to the right  marked //1 ? | | | | | | | | | |
|  | A. | 2 3 6 | | | B. | | 5 5 5 | | | |
|  | C. | 2 3 7 | | | D. | | 2 3 5 | | | |
|  | E. | 4 1 5 | | | | | | | | |
| **Question 16** | | |  | | | | | B | | |
|  | What is output by the line in the client code to the right  marked // 2 ? | | | | | | | | | |
|  | A. | 20 20 | | B. | | 12 35 | | | C. | 12 20 |
|  | D. | 35 35 | | E. | | 12 19 | | | | |
| **Question 17** | | |  | | | | | B | | |
|  | What is output by the line in the client code to the right  marked // 3 ? | | | | | | | | | |
|  | A. | -20 | | B. | | -10 | | | C. | 0 |
|  | D. | 10 | | E. | | No output due to runtime error. | | | | |
| **Question 18** | | |  | | | | | B | | |
|  | What is output by the line in the client code to the right  marked //4 ? | | | | | | | | | |
|  | A. | 5 3 7 | | | B. | | 3 0 5 | | | |
|  | C. | 0 0 0 | | | D. | | 5 3 0 | | | |
|  | E. | 5 7 3 | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question 19** | | |  | | | | | | | B | | | public int manip(int x, int y) {  int z = x \* y;  x++;  --y;  z = -(x \* y) + z;  return z;  } |
|  | What is returned by the method call manip(4, 2)? | | | | | | | | | | | |
|  | A. | 23 | | B. | | | 13 | | | | C. | 10 |
|  | D. | 5 | | E. | | | 3 | | | | | |
| **Question 20** | | |  | | | | | | | B | | | double a1 = 12.350;  double b1 = a1 % 10.1;  System.out.print(b1); |
|  | What is output by the code to the right? | | | | | | | | | | | |
|  | A. | 2.35 | | | | B. | | | 2.25 | | | |
|  | C. | 0.35 | | | | D. | | | 0.25 | | | |
|  | E. | There is no output due to a syntax error. | | | | | | | | | | |
| **Question 21** | | |  | | | | | | | B | | | String stars = "";  for(int i = 0; i <= 10; i++)  for(int j = 0; j < 20; j++)  stars += "\*";  System.out.print(stars.length()); |
|  | What is output by the code to the right? | | | | | | | | | | | |
|  | A. | 20 | | B. | | | 171 | | | | C. | 180 |
|  | D. | 200 | | E. | | | 220 | | | | | |
| **Question 22** | | |  | | | | | | | B | | | double a22 = 1.6;  int x22 = 3;  System.out.print( (int) a22 + " " +  (double) x22); |
|  | What is output by the code to the right? | | | | | | | | | | | |
|  | A. | 1 3 | | B. | | | 2 3.0 | | | | C. | 1 3.0 |
|  | D. | 2 3.6 | | E. | | | 1 2.999999999 | | | | | |
| **Question 23** | | |  | | | | | | | B | | | boolean p, q;  // code in initialize p and q  boolean r = p ^ q; |
|  | How many combinations of values of the boolean variables p and q will result in r being set to true? | | | | | | | | | | | |
|  | A. | 1 | | | | B. | | | 2 | | | |
|  | C. | 3 | | | | D. | | | 4 | | | |
|  | E. | There is no output due to a syntax error. | | | | | | | | | | |
| **Question 24** | | |  | | | | | | | B | | | Object obj = "bevo";  System.out.print(obj.equals(null));  System.out.print(obj.length()); |
|  | What is output by the code to the right? | | | | | | | | | | | |
|  | A. | false 4 | | B. | | | false 3 | | | | C. | true 3 |
|  | D. | There is no output due to a syntax error. | | | | | | | | | | |
|  | E. | There is no output due to a runtime error. | | | | | | | | | | |
| **Question 25** | | |  | | | | | | | B | | | int[] data5 = {4, 7, 2, 5, 2};  for(int i : data5)  System.out.print(i); |
|  | What is output by the code to the right? | | | | | | | | | | | |
|  | A. | 4, 7, 2, 5 | | | B. | | | 01234 | | | C. | 47252 |
|  | D. | iiiii | | | E. | | | 4, 7, 2, 5, 2 | | | | |
| **Question 26** | | |  | | | | | | | B | | | String st6 = "alamo";  st6 = st6.length() < 5 ? st6.substring(2) :  st6.substring(0,3);  System.out.print(st6); |
|  | What is output by the code to the right? | | | | | | | | | | | |
|  | A. | ala | | | | B. | | | alamo | | | |
|  | C. | amo | | | | D. | | | alam | | | |
|  | E. | There is no output due to a syntax error. | | | | | | | | | | |
| **Question 27** | | |  | | | | | | | D | | | public class Loot {  public void show() {  System.out.print("Loot");  }  public void drop() {  System.out.print("Plunk");  }  }  public class Purple extends Loot {  public void show() {  System.out.print("Shiny");  } }  // section 1 start  Loot t81 = new Loot(); Loot t82 = new Loot(); System.out.print(t81.equals(t82)); System.out.print(" " + (t81 == t82));  // section 1 stop  // section 2 start  Loot t91 = new Loot(); Purple p91 = new Purple(); t91.show(); p91.drop();  // section 2 stop  // section 3 start  Loot t31 = new Purple(); t31.show(); System.out.print(t31 instanceof Object); System.out.print(t31 instanceof Purple);  // section 3 stop |
|  | Given classes Loot and Purple to the right what is output by the following client code?  Loot.show(); Loot.drop(); | | | | | | | | | | | |
|  | A. | LootPlunk | | | | | | | | | | |
|  | B. | PlunkLoot | | | | | | | | | | |
|  | C. | LootShiny | | | | | | | | | | |
|  | D. | ShinyLoot | | | | | | | | | | |
|  | E. | There is no output due to a syntax error in the client code. | | | | | | | | | | |
| **Question 28** | | |  | | | | | | | B | | |
|  | What is output by the client code marked section 1? | | | | | | | | | | | |
|  | A. | false false | | | | B. | | | false true | | | |
|  | C. | true false | | | | D. | | | true true | | | |
|  | E. | There is no output due to a syntax error in section 1 of the client code. | | | | | | | | | | |
| **Question 29** | | |  | | | | | | | B | | |
|  | What is output by the client code marked section 2? | | | | | | | | | | | |
|  | A. | ShinyPlunk | | | | B. | | | Loot | | | |
|  | C. | LootPlunk | | | | D. | | | Shiny | | | |
|  | E. | There is no output due to a syntax error in section 2 of the client code. | | | | | | | | | | |
| **Question 30** | | |  | | | | | | | D | | |
|  | What is output by the client code marked section 3? | | | | | | | | | | | |
|  | A. | Shinytruefalse | | | | | | | | | | |
|  | B. | Loottruefalse | | | | | | | | | | |
|  | C. | Loottruetrue | | | | | | | | | | |
|  | D. | Shinytruetrue | | | | | | | | | | |
|  | E. | There is no output due to a syntax error in section 3 of the client code. | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question 31** | | |  | | | | | | D | | | int[] data = {3, 1, 3, 2};  TreeSet<Integer> ts;  ts = new TreeSet<Integer>();  HashSet<Integer> hs;  hs = new HashSet<Integer>();  for(int x31 : data) {  ts.add(x31);  hs.add(x31);  }  System.out.print(ts.equals(hs));  hs.remove(1);  hs.remove(2);  System.out.print(ts);  System.out.print(hs); | | | | |
|  | What is output by the code to the right? | | | | | | | | | | |
|  | A. | false[1, 2, 3][3] | | | | | | | | | |
|  | B. | true[3, 1, 3, 2][3, 3] | | | | | | | | | |
|  | C. | true[1, 2, 3][3] | | | | | | | | | |
|  | D. | false[3, 1, 3, 2][3, 3] | | | | | | | | | |
|  | E. | The output will vary from one run of the program to the next. | | | | | | | | | |
| **Question 32** | | |  | | | | | | D | | | public String stManip(String st) {  if(st.length() < 3)  return st + st;  else  return stManip(st.substring(1)) +  stManip(st.substring(3));  }  // client code  String st30 = "The\_Drag";  String st31 = stManip(st30);  System.out.print(st31.length()); | | | | |
|  | What is output by the code to the right? | | | | | | | | | | |
|  | A. | 15 | | | | | | | | | |
|  | B. | 30 | | | | | | | | | |
|  | C. | 34 | | | | | | | | | |
|  | D. | 68 | | | | | | | | | |
|  | E. | There is no output due to a runtime error. | | | | | | | | | |
| **Question 33** | | |  | | | | | | D | | | double a33 = 1.5;  double b33 = a33 - 1.5;  double c33 = -a33 + 1.5; System.out.print(a33 / b33); // 1 System.out.print(" " + (b33 / c33)); // 2 | | | | |
|  | What is output by the code to the right? | | | | | | | | | | |
|  | A. | Infinity Infinity | | | | | | | | | |
|  | B. | Infinity NaN | | | | | | | | | |
|  | C. | There is no output due to a syntax error. | | | | | | | | | |
|  | D. | There is no output due to a runtime error. | | | | | | | | | |
|  | E. | 2000000000 and then a runtime error occurs | | | | | | | | | |
| **Question 34** | | |  | | | | | E | | | | | | | | |
|  | Which sorting algorithm has a best case, average case, and worst case order of O(N2) for its traditional implementation? | | | | | | | | | | | | | | | |
|  | A. | merge sort | | B. | | | radix sort | | | C. | insertion sort | | D. | quicksort | E. | selection sort |
| **Question 35** | | |  | | | | | | B | | | List<Integer> list5;  list5 = new ArrayList<Integer>();  list5.add(5);  list5.add(3);  list5.add(1, 2);  list5.set(4, 2);  list5.add(0, 6);  list5.remove(2);  System.out.print(list5); | | | | |
|  | What is output by the code to the right? | | | | | | | | | | |
|  | A. | [6, 4, 3] | | | B. | [5, 3, 1, 2, 0, 6] | | | | | |
|  | C. | [6, 5, 4] | | | D. | [5, 4, 1, 2, 0, 6] | | | | | |
|  | E. | There is no output due to a runtime error. | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question 36** | | |  | | | | | | | | D | | | | |  | | | | |
|  | Which of the following Java expressions is equivalent to the formula to the right? r1, r2, and r3 are variables of type double. | | | | | | | | | | | | | | |
|  | A. | 1.0 / 1.0 / r1 + 1.0 / r2 + 1.0 / r3 | | | | | | | | | | | | | |
|  | B. | 1 / 1 / r1 + 1 / r2 + 1 / r3 | | | | | | | | | | | | | |
|  | C. | 1 / r1 + 1 / r2 + 1 / r3 | | | | | | | | | | | | | |
|  | D. | 1 / (1 / r1 + 1 / r2 + 1 / r3) | | | | | | | | | | | | | |
|  | E. | None of A through D are correct. | | | | | | | | | | | | | |
| **Question 37** | | |  | | | | | | | E | | | | | | | | | | |
|  | Assume method sample(int[] data) is O(N2) where N = data.length. When method sample is passed an array with length = 2,000 it takes 1 second for method sample to complete. If method sample is then passed an array with length = 10,000 what is the expected time it will take method sample to complete? | | | | | | | | | | | | | | | | | | | |
|  | A. | 25 seconds | | | B. | | | | 20 seconds | | | | | C. | 10 seconds | | D. | 5 seconds | E. | 1 second |
| **Question 38** | | |  | | | | | | | | B | | | | | public int[] build(String data) {  int[] result  = new int[data.length()];  final int LIMIT = data.length() - 5;  for(int r = 0; r < LIMIT; r++)  for(int c = 0; c < 5; c++)  result[r] = data.charAt(r + c);  return result;  } | | | | |
|  | What is the order (Big O) of method build to the right? N = data.length().  Pick the most restrictive correct answer. | | | | | | | | | | | | | | |
|  | A. | O(1) | | B. | | | O(logN) | | | | | C. | O(N) | | |
|  | D. | O(NlogN) | | E. | | | O(N2) | | | | | | | | |
| **Question 39** | | |  | | | | | | | | B | | | | | public class Structure<**#1**> {  private ArrayList<**#1**> con;    public Structure() {  con = new ArrayList<**#1**>();  }  public void add(**#1** e) { con.add(e); }  public **#1** get() { return con.get(0); }  public **#1** remove() {  return con.remove(0);  }  public boolean isEmpty() {  return con.size() == 0;  }  } | | | | |
|  | Which of the following can replace **#1** in the code to the right so that the Structure class compiles without error? | | | | | | | | | | | | | | |
|  | A. | E | | | | B. | | Set | | | | | | | |
|  | C. | \_1 | | | | D. | | anyType | | | | | | | |
|  | E. | All of A through D are correct. | | | | | | | | | | | | | |
| Assume **#1** has been replaced correctly. | | | | | | | | | | | | | | | |
| **Question 40** | | |  | | | | | | | | B | | | | |
|  | What type of data structure does the Structure class implement? | | | | | | | | | | | | | | |
|  | A. | min heap | | B. | | | linked list | | | | | C. | set | | |
|  | D. | queue | | E. | | | array based list | | | | | | | | |

# Standard Classes and Interfaces — Supplemental Reference

|  |  |
| --- | --- |
| **class java.lang.Object**   * boolean equals(Object other) * String toString() * int hashCode()   **interface java.lang.Comparable<T>**   * 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>**   * Integer(int value) * int intValue() * boolean equals(Object obj) * String toString() * int compareTo(Integer anotherInteger) * static int parseInt(String s)   **class java.lang.Double implements**  **Comparable<Double>**   * Double(double value) * double doubleValue() * boolean equals(Object obj) * String toString() * int compareTo(Double anotherDouble) * static double parseDouble(String s)   **class java.lang.String implements**  **Comparable<String>**   * int compareTo(String anotherString) * boolean equals(Object obj) * int length() * String substring(int begin, int end) Returns the substring starting at index begin and ending at index (end - 1). * String substring(int begin) Returns substring(from, length()). * 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 the specified index.. Returns -1 if str is not found. * charAt(int index) * int indexOf(int ch) * int indexOf(int ch, int fromIndex) * String toLowerCase() * String toUpperCase() * String[] split(String regex) * boolean matches(String regex)     **class java.util.Stack<E>**   * boolean isEmpty() * E peek() * E pop() * E push(E item)   **interface java.util.Queue<E>**   * boolean add(E e) * boolean isEmpty() * E peek() * E remove()   **class java.util.PriorityQueue<E>**   * boolean add(E e) * boolean isEmpty() * E peek() * E remove()   **interface java.util.Set<E>**   * boolean add(E e) * boolean contains(Object obj) * boolean remove(Object obj) * int size() * Iterator<E> iterator() * boolean addAll(Collection<? extends E> c) * boolean removeAll(Collection<?> c) * 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>**   * 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()   **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>**   * K getKey() * V getValue() * V setValue(V value)   **interface java.util.Iterator<E>**   * boolean hasNext() * E next() * void remove()   **interface java.util.ListIterator<E> extends**  **java.util.Iterator<E>**  Methods in addition to the Iterator methods:   * void add(E e) * void set(E e) | **class java.lang.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 java.lang.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, in b) * static int max(int a, int b) * static long round(double a) * 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>**   * boolean add(E e) * int size() * Iterator<E> iterator() * ListIterator<E> listIterator() * E get(int index) * E set(int index, E e) Replaces the element at index with the object e. * 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. * 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:   * void addFirst(E e) * void addLast(E e) * E getFirst() * E getLast() * E removeFirst() * E removeLast()   **class java.lang.Exception**   * Exception() * Exception(String message)   **class java.util.Scanner**   * Scanner(InputStream source) * boolean hasNext() * boolean hasNextInt() * boolean hasNextDouble() * String next() * int nextInt() * double nextDouble() * String nextLine() * Scanner useDelimiter(String pattern) |

# Computer Science Answer Key

# UTCS HS Contest

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **E** | 11. | **D** | 21. | **E** | 31. | **C** |
|  | **D** | 12. | **E** | 22. | **C** | 32. | **B** |
|  | **D** | 13. | **A** | 23. | **B** | 33. | **B** |
|  | **E** | 14. | **E** | 24. | **D** | 34. | **E** |
|  | **D** | 15. | **D** | 25. | **C** | 35. | **E** |
|  | **E** | 16. | **C** | 26. | **A** | 36. | **D** |
|  | **A** | 17. | **A** | 27. | **E** | 37. | **A** |
|  | **E** | 18. | **B** | 28. | **A** | 38. | **C** |
|  | **C** | 19. | **E** | 29. | **C** | 39. | **E** |
|  | **D** | 20. | **B** | 30. | **D** | 40. | **D** |

# Notes:

The clause "Choose the most restrictive correct answer." is necessary because per the formal definition of Big O, an algorithm that is O(N2) is also O(N3) , O(N4) , and so forth.

24. obj.length() causes is a syntax error. The declared type of obj is Object thus only methods from the Object class may be called without casting to String.