AP Computer Science Name _____

2004 Final Exam Part I Date _____ Block _____

Directions: Clearly and precisely circle one selection for each problem. This is a 50 minute exam. Marshal your time!

1. Mina is five years older than her brother Joe. Three years from now Mina's age will be twice Joe's age. How old are Mina and Joe now? A student wrote the following program to solve this problem:

```
public class AgeFinder
{
   public static void main(String[] args)
   {
      for(int a = 1; a <= 100; a++)
         for(int b = 1; b <= 100; b++)
            if( < condition > )
              System.out.println("Mina is " + a + " and Ben is " + b);
   }
}
```

Which of the following expressions should replace `< condition >` in the student's program so that it displays the correct solution to the problem?

a. `(a==b-5) && (a-3==2*(b-3))`
b. `a==b+5 && a+3==2*b+6`
c. `(a==(b+5)) && ((a+3)==(2*b+3))`
d. `a==(b-5) && (2*a-3) == (b-3)`
e. [none of the above]

2. The Binary Search algorithm normally finds a value in an array sorted in ascending order. Suppose that by mistake the algorithm is used on an unsorted array with the following seven elements:

1 3 2 5 13 8 21

Which of the following target values will NOT be found?

a. 1
b. 3
c. 5
d. 13
e. 21

3. Let us call an array "oscillating" if its values alternate going up and down, as follows: `a[i-1] < a[i]` and `a [i] > a[i+1]` for any odd `i, 0 < i < n-1`, where `n` is the number of elements in `a`. What is the "big-O" for an optimal algorithm that determines the minimum value in an oscillating array of length `n`? The median value?

	Minimum	Median
a.	$O(1)$	$O(1)$
b.	$O(n/2)$	$O(1)$
c.	$O(n)$	$O(n)$
d.	$O(n)$	$O(n \log n)$
e.	$O(n)$	$O(n^2)$

4. The following version of Selection Sort is supposed to sort an array in ascending order. For better performance it tries to tackle the array from both ends simultaneously:

```
public void sort(int a[])
{
   int left = 0, right = a.length-1;
   int k;

   while(left < right)
   {
      for(k = left+1; k < right; k++)
      {
         if( a[k] < a[left] )
            swap(a, k, left);
         else if( a[k] > a[right] )
            swap(a, k, right);
      }
      left++;
      right--;
   }
}
```

`swap(a, i, j)` correctly swaps `a[i]` and `a[j]`. This code has a bug, though. Which of the following changes would assure that the method sorts all arrays correctly?

I. Remove the `else` in: `else if(a[k] > a[right])…`

II. Replace: `for(k=left+1; k<right; k++)` with `for(k=left; k<=right; k++)`

III. Add `if(a[left] > a[right]) swap(a, left, right)` at the beginning of the `while` loop (before the `for` loop).

a. I only
b. II only
c. III only
d. I or II
e. II or III

5. What is the output of the following code segment?

```
Map m = new TreeMap();
m.put("La", "La");
m.put("La-La", "La");
m.put("La-La-La", "Ye-Ye");
Iterator it = m.keySet().iterator();
while(it.hasNext())
   System.out.print(m.get(it.next()) + " ");
```

a. `La Ye-Ye`
b. `La La Ye-Ye`
c. `La La-La-La`
d. `La La La-La-La Ye-Ye`
e. `La La La-La La La-La-La Ye-Ye`

Questions 6-10 are based on the following classes:

```
public class Person implements Comparable
{
   private String name;

   public Person(String name)   {  this.name = name;  }

   public String getName()  {  return name;  }

   public boolean equals(Object other)
   {
      return other != null && name.equals(((Person)other).name);
   }

   public int compareTo(Object other)
   {
      return name.compareTo(((Person)other).name);
   }

   public int hashCode()  {  return name.hashCode();  }
}


public class SoccerPlayer extends Person
{
   private int numGoals;

   public SoccerPlayer(String name, int n)
   {
      super(name);
      numGoals=n;
   }

   public int getNumGoals()  {  return numGoals;  }

   public void score()  {  numGoals++;  }

   public int compareTo(SoccerPlayer other)
   {
      return getNumGoals() – other.getNumGoals();
   }

   public String toString()
   {
      return getName() + "/" + getNumgoals();
   }
}
```

6. Which of the following declarations is invalid:

a. `Person p = new Person("Mia Hamm");`
b. `SoccerPlayer p = new Person("Mia Hamm");`
c. `Comparable p = new Person("Mia Hamm");`
d. `Person p = new SoccerPlayer("Lori Fair", 0);`
e. `Comparable p = new SoccerPlayer("Lori Fair", 0);`

7. What is the result of the following code?

```
Person players[]={new SoccerPlayer("Mia Hamm",7), new SoccerPlayer("Lori Fair",6)};
System.out.println(players[0].compareTo((SoccerPlayer)players[1]));  // Line ***
```

a. Syntax error in the class `Person:other.name` is not accessible
b. Syntax error in the class `SoccerPlayer:compareTo` is redefined
c. `ClassCastException` on Line***
d. Compiles and runs with no errors; displays 1
e. Compiles and runs with no errors; displays 2

8. Suppose the `Person` and `SoccerPlayer` classes are changed as follows: `other.name` is replaced with `other.getName()` and `compareTo` in `SoccerPlayer` is renamed into `compareGoals`. What will be the result of the following code segment?

```
SoccerPlayer mia = new SoccerPlayer("Mia Hamm",6);
SoccerPlayer lori = new SoccerPlayer("Lori Fair",5);
Set team = new HashSet();
team.add(mia);
team.add(lori);
lori.score();
team.add(lori);
Iterator iter = team.iterator();          // Line ***
while(iter.hasNext())
   System.out.print(iter.next() + " ");
```

a. `Lori Fair/5 Mia Hamm/6`
b. `Lori Fair/6 Mia Hamm/6`
c. `Mia Hamm/6 Lori Fair/5 Lori Fair/6`
d. `Lori Fair/5 Lori Fair/6 Mia Hamm/6`
e. Syntax error on Line ***

Questions 9 and 10 are concerned with a class `SoccerTeam` that represents a team of soccer players:

```
public class SoccerTeam
{
   private SoccerPlayer[] players;
   private ArrayList mvps;  //holds all the players who scored the same highest number of goals
   public void score(int k)
   {
      players[k].score();                                          //Line *1*
      int goals=players[k].getNumGoals();
      int maxGoals=((SoccerPlayer)mvps.get(0)).getNumGoals();      //Line *2*
      if(goals >=maxGoals)
      {
         if(goals == maxGoals)                                     //Line *3*
            mvps.add(players[k]);
         else
         {
         < missing statement > //mvps is left with only one player in it, players[k]:
         }
      }
   }
   < constructors and other methods not shown >
}
```

9. `SoccerTeam`'s `score` method is intended to update the number of scored goals for a given player on the team and update the list of "most valuable players" (all of whom have the same score, the highest on the team). If the player's new score is higher than the old best, the `mvps` list is updated to contain only that one player. However, the `score` method has an error. Which of the following actions would correct that error?

I. Move Line*2* before Line*1*
II. Replace Line *3* with `if(goals==maxGoals && players[k] != mvps.get(0))`
III. Replace Line *3* with `if(goals==maxGoals && !mvps.contains(players[k]))`

a. I only
b. II only
c. I and II
d. II and III
e. I, II, and III

10. Which of the following would be an appropriate replacement for < missing statements > in `SoccerTeam`'s `score` method?

a.
```
mvps.set(0, players[k]);
```
b.
```
mvps.resize(0);
mvps.add(players[k]);
```
c.
```
mvps = new ArrayList();
mvps.add(players[k]);
```
d.
```
mvps = new ArrayList(1);
mvps.set(0, players[k]);
```
e.
```
delete mvps;
mvps = new ArrayList();
mvps.add(players[k]);
```

11. What does the following code display?

```
String expr = "(a+b)/(2*(a-b))";
Stack stk = new ArrayStack();
int i,k;

for(k=0; k<expr.length(); k++)
{
  if(expr.charAt(k) == '(' )
  {
    stk.push(new Integer(k+1));
  }
  else if(expr.charAt(k) == ')' )
  {
    i = ((Integer)stk.pop()).intValue();
    System.out.println(expr.substring(i,k));
  }
}
```

a. `(2 * (a-b))`
   `(a-b)`
   `(a+b)`
b. `(a-b)`
   `(2 *(a-b))`
   `(a+b)`
c. `a+b`
   `a-b`
   `2 * (a-b)`
d. `a+b`
   `a-b`
   `2 * a-b`
e. `a+b)`
   `a-b)`
   `2 * (a-b))`

12. Suppose `ListNode p1, p2` initially refer to two nodes in the same circular linked list. Under what condition does the following loop terminate?

```
do
{
  p1 = p1.getNext();
  p2 = p2.getNext().getNext();
}while(p1 != p2);
```

a. Always
b. If and only if `p1==p2.getNext()`
c. If and only if the total number of node in the list is even
d. If and only if the number of nodes from `p2` to `p1` (excluding both ends of this segment of the list) is even
e. If and only if the list contains two nodes with the same `info`

13. What is the contents of the stack stk(its elements listed starting from the top) after the following code is executed?

```
Stack stk = new ArrayStack();
Stack stk1 = new ArrayStack();
Stack stk2 = new ArrayStack();

int n;
Integer obj;
for(n=1; n<=6; n++)
  stk.push(new Integer(n));

while(!stk.isEmpty())
{
  obj = (Integer)stk.pop();
  n = obj.intValue();
  if(n%2 !=0)
    stk1.push(obj);
  else
    stk2.push(obj);
}
while(!stk1.isEmpty())
  stk.push(stk1.pop());
while(!stk2.isEmpty())
  stk.push(stk2.pop());
```

a. 1,2,3,4,5,6
b. 6,5,4,3,2,1
c. 1,3,5,2,4,6
d. 2,4,6,1,3,5
e. None of the above

14. Suppose an *n* by *n* matrix of "black" and "white" pixels (e.g., 1s and 0s) represents a picture of a black blob that fills the southeastern corner of the picture. The blob's boundary extends in a generally southwest-to-northeast direction. All the pixels below and to the right of any black pixel are black. For example:

0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	1
0	0	0	1	1	1	1
0	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

What is the worst-case "big-O", in terms of *n*, for the total number of integer additions plus pixel comparisons in an optimal algorithm that determines the area of a blob (the number of black pixels in the blob)?

a. $O(\log n)$
b. $O((\log n)^2)$
c. $O(n)$
d. $O(n \log n)$
e. $O(n^2)$

15. Which of the following code segments will compile with no errors and produce an alphabetical list of all three pets:

Lady – Golden retriever
Spot – Dalmatian
Tramp – Labrador retriever

I.
```
Set set = new TreeSet();
set.add(Lady);
set.add(Spot);
set.add(Tramp);
Iterator iter = set.iterator();
while(iter.hasNext())
  System.out.println(iter.next());
```

II.
```
Map map = new TreeMap();
map.put(Lady.getName(),Lady);
map.put(Spot.getName(),Spot);
map.put(Tramp.getName(),Tramp);
Iterator iter = map.values().iterator();
while(iter.hasNext())
  System.out.println(iter.next());
```

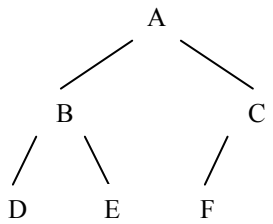
III.
```
Map map = new HashMap();
map.put(Lady.getName(),Lady);
map.put(Spot.getName(),Spot);
map.put(Tramp.getName(),Tramp);
Iterator iter = map.keyset().iterator();
while(iter.hasNext())
  System.out.println(map.get(iter.next()));
```

a. I only
b. II only
c. I and II
d. II and III
e. I, II, and III

16. Consider the following method that builds a linked list from a binary tree:

```
public ListNode treeToList(TreeNode root)
{
  if(root == null)
    return null;
  else if(Math.random()<0.5)
    return new ListNode(root.getValue(), treeToList(root.getLeft()));
  else
    return new ListNode(root.getValue(), treeToList(root.getRight()));
}
```

if `root` refers to the tree



Which of the following lists could possibly result from `treeToList(root)` ?

I. A, B, D

II. A, B, C, D

III. A, C, F

a. I only
b. II only
c. I and II
d. I and III
e. I, II, and III

17. Consider the following method that builds a binary tree from a queue:

```
public TreeNode queueToTree(ListQueue q)
// precondition: q holds Comparable objects
{
  if(q.isEmpty())
    return null;

  ListQueue q1 = new ListQueue();
  ListQueue q2 = new ListQueue();
  Comparable x,y;
  x = (Comparable)q.dequeue();
  while(!q.isEmpty())
  {
    y = (Comparable)q.dequeue();
    if(y.compareTo(x) < 0)
      q1.enqueue(y);
    else
      q2.enqueue(y);
  }
  return new TreeNode(x, queueToTree(q1), queueToTree(q2));
}
```

If `q` contains letters A, P, R, I, C, O, T (represented by `Character` or `String` objects, in this order, from front to rear), what is the result of inorder traversal (left-root-right) of the tree returned by `queueToTree(q)`?

a. A, C, I, O, P, R, T
b. A, P, R, T, I, C, O
c. A, P, R, C, I, O, T
d. A, C, O, I, T, R, P
e. A, P, I, C, O, R, T

18. Let n be the number of fish in the environment. Assume that UnboundedEnv is implemented efficiently. What would be the average-case running time of the method isEmpty in UnboundedEnv if the environment was represented as a HashMap?

a. $O(1)$
b. $O(\log n)$
c. $O(n)$
d. $O(n^2)$
e. The answer depends on the number of rows and columns occupied by fish.

19. Consider the following method `eval`:

```
String eval(Queue q)
{
   Stack stk = new ArrayStack();
   String s = "";
   while(!q.isEmpty())
   {
     s = (String)q.dequeue();
     if(!s.equals("+"));
       stk.push(s);
     else
     {
       String s1 = "", s2="";
       if(!stk.isEmpty())
         s2 = (String)stk.pop();
       if(!stk.isEmpty())
         s1 = (String)stk.pop();
       stk.push(s1 + s2);
     }
   }
   if(!stk.isEmpty())
     s = (String)stk.pop();
   return s;
}
```

The program reads strings (which may hold single characters), separated by spaces, one by one from the input line and puts them into a queue `q`. Then it displays the string returned by `eval(q)`. For which of the following input lines is the output `HELLO`?

front

a. H + E + L + L + O
b. H E L + + L O + +
c. H E + + L L + O +
d. + + + + O L L E H
e. None of the above.

20. Which of the following statements about objects and classes is true?

a. A `final` variable in a class must be declared as `private`.
b. Multiple object references cannot contain references to the same object.
c. A client program can creates only one object of a class.
d. A `static` method is a method that operates on an object that has been instantiated.
e. Every object belongs to a class.

21. The method below implements a simplified square cipher:

```
public char[][] encrypt(char[][] key, String msg)
{
  int i, j, n = key.length, k = 0;
  char[][] result = new char[n][n];    //fills with spaces

  for(i = 0; i<n; i++)
    for(j = 0; j<n; j++)
      if(key[i][j] =='X')
        result[i][j] = msg.charAt(k++);

  for(i = 0; i<n; i++)
    for(j = 0; j<n; j++)
      if(key[i][j] =='X')
        result[n-i-1][n-j-1] = msg.charAt(k++);

  return result;
}
```

If `key` is a 4 X 4 matrix

.	X	.	.
X	.	.	X
.	X	X	.
X	X	.	X

and msg is "RANSOM RECEIVED ", which of the following matrices is returned from encrypt(key, msg)?

a.

R	E	A	N
C	S	O	E
M	I	V	R
D		E	

b.

	R	D	E
A	V	I	N
E	S	O	C
M		E	R

c.

	R	V	C
A	E	I	N
D	S	O	E
M		E	R

d.

R	D	S	R
	N	O	
A	E	V	E
I	E	M	C

e.

R	C	A	N
E	S	O	I
M	V	R	E
D		E	E

22. If the 2-D array theGrid in the BoundedEnv class was replaced by a linked list of Locatable objects, which of the following method's running time would improve if the environment was sparsely populated?

a. isValid
b. numObjects
c. allObjects
d. objectAt
e. add

23. Assume that `fsh` has been defined and initialized as a `Fish` object in a client class that contains the following code segment.

```
int leftCounter = 0;
for(int k = 0; k<100; k++)
{
  Direction dir = fsh.direction();
  Direction dirLeft = dir.toLeft();

  fsh.act();
  if( /* condition */ )
    leftCounter++;
}
```

Which of the following could be used to replace `/* condition */` so that the variable `leftCounter` accurately stores the number of times that `fsh` turned to the left?

a. `dir.equals(dirLeft)`
b. `dir.equals(fsh.direction())`
c. `dirLeft.equals(dir)`
d. `dirLeft.equals(fsh.direction())`
e. `(fsh.direction()).equals(new Direction(dir))`

Questions 24 and 25 refer to the following segment of code.

```
String bookName = "Computing Concepts in Java";
String newWord =  bookName.substring(10,18) +
                  bookName.substring(6,8) + bookName.substring(0,9);
```

24. After the code is executed, what is the `String` that `newWord` references?

a. " ConcepttinComputin"
b. "Concepts ingComputing "
c. Concepts in Computing"
d. "ConceptsinComputing"
e. An error message is generated.

25. The statement `System.out.println(bookName.indexOf("Cat"));` prints.

a. *–1*
b. *0*
c. *null*
d. *1*
e. An error message is generated

26. A hash function computes an integer value from an object. The goal of a good hash function is to:

a. determine the size of the hash table.
b. provide space for the object that is to be inserted.
c. provide a method of dealing with collisions
d. provide a key for efficiently sorting the objects in the hash
e. provide an integer value for the object so that the objects are uniformly distributed in the hash table.

27. Consider the following algorithms to accomplish the task of removing duplicates from an array list `list`.

I. For each element in `list` (x_i), look at x_i. Now look at each element in `list` and count how many times x_i occurs in `list`. If the count is larger than 1, remove x_i.

II. Sort the array list, with an efficient sort algorithm. For each x_i in `list`, look at its next neighbor to decide whether it is present more than once. If it is, remove x_i.

III. Sort the array list, `list`, with an efficient algorithm. Traverse the array list looking at each x_i in `list`. Whenever the element after x_i is strictly larger than x_i, append x_i to a second (initially empty) array list, `list2`. Find the difference, `diff`, in sizes of `list` and `list2`. Beginning with the last element of `list`, delete `diff` elements from `list`. Then copy the elements of `list2` back into `list`.

Of the following, which best describes the running time of these algorithms?

a. I, II, and III are all $O(n^2)$.
b. I and II are $O(n^2)$, III is $O(n \log(n))$.
c. I is $O(n^2)$, II and III are $O(n \log(n))$.
d. I and III are $O(n^2)$, II is $O(n \log(n))$.
e. I, II, and III are $O(n \log(n))$.

28. The quicksort algorithm is to be used to sort an array of integers in decreasing order. Which of the cases below describes the worst case for the quicksort algorithm?

I. The original array is in increasing order.
II. The original array is in decreasing order.
III. The original array is in random order.

a. I only
b. II only
c. III only
d. I and II only
e. I, II, and III all take the same number of comparisons

29. Consider the following definition of a heap.

A *heap* is a binary tree with the following properties:

- It is complete or almost complete, which means that every level of the tree is completely filled, except maybe at the bottom level. If the bottom level is not filled, the nodes are in the leftmost positions.
- The object stored at each node is at least as large as the values stored in its children.
- Adding an element to the heap maintains the heap properties

The following `Integers` are added to an initially empty heap in the order they are listed. The heap properties are maintained with each insertion.

9 25 32 90 15 4 23

If the heap is represented by a binary tree and an inorder traversal is done on this binary tree, the resulting order of the visited `Integers` is:

a. 9 32 15 90 4 25 23
b. 4 9 15 23 25 32 90
c. 90 32 25 23 15 9 4
d. 90 32 25 9 15 4 23
e. 9 15 32 4 23 25 90

30. Which of the following statements about searching is **not** true?

a. The linear search examines all values in an array until it finds a match or until it reaches the end.
b. A binary search is generally more efficient than a linear search.
c. A binary search can only be used to search for an item in a sorted array.
d. A sequential search generally takes more comparisons than a binary search.
e. A binary search is always faster than a sequential search.