Questions 1. [2 marks]

Any two of the following four would be acceptable:

- The concept of block structure was introduced.
- Two different means of passing parameters to subprograms were allowed: pass by value and pass by name.
- Procedures were allowed to be recursive.
- Stack-dynamic arrays were allowed.

Question 2. [2 marks]

1) Has proven to be highly inefficient
2) Has shown to be an effective method for few small areas of application like some kind of database management system and some areas of AI.

Question 3. [4 marks]

Any two of the following nine would be acceptable:

- Has both classes and primitive types, which are not objects based on classes, as does C++.
- Objects are accessed through reference variables, but the primitive types are value types.
- Java arrays are instances of a predefined class, whereas in C++ are not.
- Java doesn’t have pointers, instead it has reference types that provide some of the capabilities of pointers. References and pointers are almost alike, but pointers point to memory locations, while references point to objects, which makes arithmetic between references nonsense.
- Unlike C/C++ Java has Boolean primitive type used mainly for control expressions.
- Java does not support procedural programming, there is no construct in Java that is called a function or a sub program; methods can only be called through a class or object.
- C++ supports multiple inheritances directly in its class definitions. Java supports only single inheritance, though you can still gain multiple inheritances by the use of its interface construct.
- Java includes a relatively simple form of concurrency control through its synchronize modifier, which can appear in methods and blocks. In java it is relatively easy to create threads.
- Addition of the Garbage Collector.
Question 4. [4 marks]

a) Language design time
   e.g., bind operator ”=” to assignment

b) Compile time
   e.g., bind x to int in
   int x; // in C or Java

c) Load time
   e.g., bind x to a memory cell in
   static int x; // in C

d) Run time
   e.g., bind x to a memory cell in
   int x; // in C

Question 5. [2 marks]

Too much access to data

Question 6. [2 marks]

“Float” is not an ordinal type because it’s not possible to map all of its members easily to some positive integer numbers.

Question 7. [5 marks]

addressOf(M[i,j])=addressOf(M[lb2, lb1])
+( (j-lb1) * (ub2-lb2+1)
 + (i-lb2) ) * sizeof(REAL)

Question 8. [2 marks]

![Diagram of 3D grid with dimensions labeled]
Question 9. [24 marks]

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Visible Variables</th>
<th>Where Declared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence (a) with static scoping</td>
<td>A,B,C</td>
<td>Sub3 Main</td>
</tr>
<tr>
<td>Sequence (a) with dynamic scoping</td>
<td>A,B,E,D,C</td>
<td>Sub3 Sub2 Sub1 Main</td>
</tr>
<tr>
<td>Sequence (b) with static scoping</td>
<td>A,B,C</td>
<td>Sub3 Main</td>
</tr>
<tr>
<td>Sequence (b) with dynamic scoping</td>
<td>A,B,C</td>
<td>Sub3 Main</td>
</tr>
<tr>
<td>Sequence (c) with static scoping</td>
<td>A,B,E,D,C</td>
<td>Sub2 Sub1 Main</td>
</tr>
<tr>
<td>Sequence (c) with dynamic scoping</td>
<td>A,B,E,D,C</td>
<td>Sub2 Sub1 Main</td>
</tr>
</tbody>
</table>

Question 10. [4 marks]

**Advantages:** accuracy

**Disadvantages:** Any of the following would be accepted
- Limited range,
- waste memory

Question 11. [4 marks]

<table>
<thead>
<tr>
<th>Record location</th>
<th>0</th>
<th>flag</th>
<th>bool</th>
<th>4</th>
<th>value</th>
<th>double</th>
<th>5</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 12. [5 marks]

a) [2 marks]

b) [3 marks]

Constrained variant record
(Or Descriminated Union)

Question 13. [4 marks]

Is a dangling pointer or a lost-heap-dynamic-variable (or both) anywhere in the following C pseudo-code?

NO

Question 14. [6 marks]

Two different trees for the same sentential form of:  \( f \ f \ <S> \ e \ <S> \)