StudentID# : (   ) , Name : (   )

※ You may answer either in English or Korean language. As an exception, Problem 1 should be answered only in English.

1. (24points) Fill out the blanks (a)~(k) with the most appropriate English words.

● NUMA is a computer memory design used in multiprocessing,
  where the memory access time depends on (a.   ).

● Parallelism overheads include
  cost of (b.   ),
  cost of (c.   ),
  cost of (d.   ), and
  (e.   ).

● Amdahl's law: even if the (f.   ) part speeds up perfectly, performance is (g.   )
  by the (h.   ) part.

● Dynamic load balancing means that load balancing occurs at (i.   ) time. Compared to static load balancing, dynamic
  load balancing is more suitable for codes where work is (j   ) and in (k.   ) multicore/distributed
  systems.

● Fine-grain parallelism can help reduce overheads due to (l.   ).

2. (14 points) In Java multi-threaded programming,
(a) What does **wait()** method (function) do? Explain with full details.
  (Your explanation should include what happens after **wait()** is called.)
(   )

(b) What does **notify()** method (function) do? Explain with full details.
  (Your explanation should include what happens after **notify()** is called.)
(   )

3. (10 points) Fill out following blanks (a) ~ (e) with appropriate words.

< system view of a single-threaded and multithreaded process >

4. (12 points) Determine whether each of following statements regarding NUMA memory design is True or False. If you are not
sure, you may not give your answer, because if your answer is wrong, penalty (score subtraction: -4points) is applied.
(a) In NUMA, memories are physically partitioned.   (True / False)
(b) In NUMA, all processors can access all memory as global address space.   (True / False)
(c) In UMA, placement of data significantly affects performance.   (True / False)

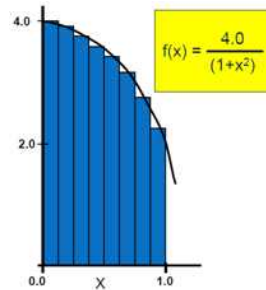5.(40 points) Following multi-threaded java code efficiently computes and displays the result of numerical integration $\int_0^1 \frac{4.0}{(1+x^2)} dx$. Fill out empty boxes below with appropriate java codes.

--------------------------------< source code : ex2.java >--

```java
class IntThread extends Thread {
  int my_id; // fields for communicating inputs
  int num_steps;
  int num_threads;
  double sum;

  IntThread(int id, int numSteps, int numThreads) {
    my_id=id; num_steps=numSteps; num_threads=numThreads;
    sum=0.0;
  }
  public void run() {
    double x, step;
    int i, i_start, i_end;




    }
  public double getSum() { return sum; }
}


class ex3 {
  private static final int NUM_THREAD = 4;      // number of threads to use
  private static final int NUM_STEP = 10000000; // number of steps (rectangles to create)

  public static void main(String[] args) {
    int i;
    double sum=0.0;




    System.out.println("integration result=" + sum +"\n");
  }
}
```

Output result :
integration result=3.1415926535896697