Methods Test 1

1. In baseball a player’s batting average is calculated by dividing the number of hits by the number of times the player has batted. Assume the following class has been defined.

public class Baseball

{

/\* precondition: atBats > 0

\* postcondition: a batting average has been computed

\* @ param hits the number of hits

\* @ param atBats the number of atBats

\* @ return the batting average

\*/

public static double **average**(int hits, int atBats)

{

// implementation not shown

}

// other methods not shown

}

If a baseball player has batted 300 times and has gotten 300 hits which of the following lines of   
 code will correctly calculate his batting average.

1. average(100, 300);
2. player.average(100, 300);
3. Baseball.average(100, 300);
4. baseball.average(100, 300);
5. The area of a rectangle is calculated by multiplying the length times the width. Assume the following class has been defined.

public class Rectangle

{

\* postcondition: the area has been computed

\* @ return the area

\*/

public double **area**()

{

// implementation not shown

}

// other methods not shown

}

Given the following definition:

Rectangle rect = new Rectangle();

Which of the following lines of code will correctly calculate the area of a rectangle?

1. Rectangle.area();
2. rect.area();
3. area();
4. rect.area(length, width);
5. Look at the following method declaration

public \_\_\_\_\_\_\_\_\_\_\_\_\_\_ sum(double num1, double num2)  
{  
 return num1 + num2;  
}

Which of the following can be inserted into the blank so that the method will correctly  
 calculate and return the sum of num1 and num2?

1. void
2. static
3. int
4. double
5. Look at the following method declaration

public void play(int a)

Which of the following will **not** correctly overload the method **play**?

1. public int play(int a)
2. public void play()
3. public void play(int a, int b)
4. public void play(double a)
5. Look at the following code segment.

public class Demo {  
 public static void go(int num) {  
 num = num + 100;  
 }

public static void main(String[] args) {  
 int number = 25;  
 go(number);  
 System.out.println(number);  
 }

}

What would be output by the code above?

1. 25
2. 100
3. 125
4. No output due to a run-time error
5. Look at the following class definition.

public class myClass {  
 private int num;  
  
 public myClass(int n) {  
 num = n;  
 }  
  
 public void setNum(int n) {  
 num = n;  
 }  
}

Which of the following will **not** correctly assign a value of 5 to the instance variable **num**?

1. MyClass obj = new MyClass(5);
2. obj.setNum(5);
3. num = 5;
4. int n = 5;

obj.setNum(n);

1. No output due to a run-time error