Do NOT write on this exam

Computer Science A
Section I

Time – 1 hour and 30 minutes
Number of questions – 40
Percent of total grade – 50

Semester 1 EOC Review

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratch work. Then decide which is the best of the choices given and fill in the corresponding box on the student answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.

Questions 1 and 2. Refer to the following code block.

```
public class Address
{
   private String street;
   private String city;
   private String state;
   private String zipCode;

   public Address(String aStreet, String aCity, String aState, int aZipCode)
   { /* implementation not shown */ }

   public String getStreet()
   { /* implementation not shown */ }

   public String getCity()
   { /* implementation not shown */ }

   public String getState()
   { /* implementation not shown */ }

   public String getZipCode()
   { /* implementation not shown */ }

   //Other methods are not shown
}
```

```
public class Customer
{
   private String name;
   private String phone;
   private Address address;
   private int ID;

   public Customer(String aName, String aPhone, Address a, int anID)
   { /* implementation not shown */ }

   public Address getAddress()
   { /* implementation not shown */ }

   public String getName()
   { /* implementation not shown */ }

   public String getPhone()
   { /* implementation not shown */ }

   public int getID()
   { /* implementation not shown */ }


   //Other methods are not shown
}
```

1. Which of the following correctly creates an Address object a?

I. `Address a = new Address("125 Bismark St", "Pleasantville", "NY", 14850);`

II. `Customer c = new Customer("Jack Spratt", "747-1674", a, 7008);`

III.
```
Customer c = new Customer("Jack Spratt", "747-1674",
new Address("125 Bismark St", "Pleasantville", "NY", 14850), 7008);
```

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) **I and III only**

2. Which of the following correctly creates a `Customer` object c?

I. `Address a = new Address("125 Bismark St", "Pleasantville", "NY", 14850);`

II. `Customer c = new Customer("Jack Spratt", "747-1674", address, 7008);`

III.
```
Customer c = new Customer("Jack Spratt", "747-1674",
new Address("125 Bismark St", "Pleasantville", "NY", 14850), 7008);
```

(A) I only

(B) II only

(C) **III only**

(D) I and II only

(E) I and III only

3. Consider the following method,

```
public static int mystery(int[] arr)
{
  int x = 0;
  for(int k = 0; k < arr.length; k = k + 2)
        x = x + arr[k];
        return x;
}
```

Assume that the array `nums` has been declared and initialized as follows,

```
int[] nums = {3, 6, 1, 0, 1, 4, 2};
```

What value will be returned as a result of the call `mystery(nums)` ?

(A) 5

(B) 6

(C) **7**

(D) 10

(E) 17

Questions 4 and 5. Refer to the following code block.

```
public class SomeClass
{
   private int myA;
   private int myB;
   private int myC;

   //Constructor(s) not shown

   public int getA()
   { return myA; }

   public void setB( int value )
   { myB = value; }

}
```

4. Which of the following changes to SomeClass will allow other classes to access but not modify the value of myC?

(A) Make `myC` public

(B) **Include the method:**
```
public int getC()
{ return myC; }
```

(C) Include the method:
```
private int getC()
{ return myC; }
```

(D) Include the method:
```
public void getC(int x)
{ x = myC; }
```

(E) Include the method:
```
private void getC(int x)
{ x = myC; }
```

5. The following declaration appears in another class

```
SomeClass obj = new SomeClass();
```

Which of the following code segments will compile without error?

(A) `int x = obj.getA();`

(B)
```
int x;
obj.getA(x);
```

(C) `int x = obj.myA`

(D) `int x = SomeClass.getA();`

(E) `int x = getA(obj);`

6. Consider the following code segment

```
int x = 7;
int y = 3;

  if((x < 10) && (y < 0))
        System.out.println(“Value is: “ + x * y);
  else
        System.out.println(“Value is: “ + x / y);
```

What is printed as a result of executing the code segment?

(A) `Value is : 21`

(B) `Value is: 2.3333333`

(C) **`Value is: 2`**

(D) `Value is: 0`

(E) `Value is: 1`

7. Consider the following method that is intended to determine if the double values d1 and d2 are close enough to be considered equal. For example, given a tolerance of 0.001, the values 54.32271 and 54.32294 would be considered equal

```
/** @return true of d1 and d2 are within the specified tolerance,
 *          false otherwise
 */
public boolean almostEqual(double d1, double d2, double tolerance)
{  /* missing code */  }
```

Which of the following should replace `/* missing code */` so that `almostEqual` will work as intended?

(A) `return (d1 – d2) <= tolerance;`

(B) `return ((d1 + d2) / 2) <= tolerance;`

(C) `return (d1 – d2) >= tolerance;`

(D) `return ((d1 + d2) / 2) >= tolerance;`

(E) **`return Math.abs(d1 – d2) <= tolerance;`**

8. Consider the following class declaration

```
public class Person
{
            private String myName;
            private int myYearOfBirth;

            public Person(String name, int yearOfBirth)
            {
                 myName = name;
                 myYearOfBirth = yearOfBirth;
            }

    public String getName()
    { return myName; }

    public void setName(String name)
    { myName = name; }

    //There may be instance variables constructors, and methods that are not shown
 }
```

Assume that the following declaration has been made,

```
Person student = new Person(“Thomas”, 2015);
```

Which of the following statements is the most appropriate for changing the name of student from “Thomas” to “Tom”?

(A) `student = new Person(“Tom”, 2015);`

(B) `student.myName = “Tom”;`

(C) `student.getName(“Tom”);`

(D) **`student.setName(“Tom”);`**

(E) `Person.setName(“Tom”);`

9. Consider the following class declaration

```
public class Student
{
            private String myName;
            private int myAge;

   public Student()
   {
       /* implementation not shown */
   }

   public Student(String name, int age)
   {
       /* implementation not shown */
   }

   //No other constructors
}
```

Which of the following declarations will compile without error?

I. `Student a = new Student();`

II. `Student b = new Student("Juan", 15);`

III. `Student c = new Student("Juan", "15");`

(A) I only

(B) II only

(C) **I and II**

(D) I and III

(E) I, II, and III

10. The following shuffling method is used to shuffle an array `arr` of `int` values. The method assumes the existence of a `swap` method, where `swap(arr, i, j)` interchanges the element `arr[i]` and `arr[j]`.

```
public static void shuffle (int[] arr)
{
   for(int k = arr.length - 1; k > 0; k--)
   {
            int randIndex = (int)  (Math.random() * (k + 1));
            swap(arr, k, randIndex);
   }
}
```

Suppose the initial state of `arr` is `1 2 3 4 5`, and when the method is executed the values generated for randIndex are `3, 2, 0, 1`, in that order. What will be the final state of `arr`?

(A) 5 4 1 3 2

(B) 1 2 5 3 4

(C) 4 5 1 3 2

(D) 2 5 1 3 4

(E) **5 2 1 3 4**

11. Consider the following method that is intended to return the sum of the elements in the array `key`.

```
public static int sumArray (int[] key)
{
      int sum = 0;
      for(int i = 1; i <= key.length; i++)
      {
            /* missing code */
      }
      return sum;
}
```

Which of the following statements should be used to replace `/* missing code */` so that `sumArray` will work as intended?

(A) `sum = key[i];`

(B) **`sum += key[i – 1];`**

(C) `sum += key[i];`

(D) `sum += sum + key[i – 1];`

(E) `sum += sum + key[i];`

12. Consider the following method.

```
public String mystery(String input)
{
      String output = “”;
      for(int k = 1; k <= input.length(); k = k + 2)
      {
            output += input.substring(k, k + 1);
      }
      return output;
}
```

What is returned as a result of the call `mystery(“computer”)`?

(A) computer

(B) cmue

(C) **optr**

(D) ompute

(E) Nothing is returned because an `IndexOutOfBoundsException` is thrown

13. Consider the following code segment

```
int[] arr = {7, 2, 5, 3, 0, 10};
for(int k = 0; k < arr.length – 1; k++)
{
      if(arr[k] > arr[k + 1])
            System.out.print(k + " " + arr[k] + " ")
}
```

What will be printed as a result of executing the code segment?

(A) 0 2 2 3 3 0

(B) **0 7 2 5 3 3**

(C) 0 7 2 5 5 10

(D) 1 7 3 5 4 3

(E) 7 2 5 3 3 0

14. Consider the following method, `isSorted`, which is intended to return `true` if an array of integers is sorted in nondecreasing order and to return false otherwise.

```
/** @param data an array of integers
 *  @return true if the values in the array appear in sorted (nondecreasing) order
 */
public static boolean isSorted(int[] data)
{
      /* missing code */
}
```

Which of the following can be used to replace `/* missing code */` so that `isSorted` will work as intended?

I.
```
for(int k = 1; k < data.length; k++)
{
     if(data[k – 1] > data[k])
          return false;
}
return true;
```

II.
```
for(int k = 0; k < data.length ; k++)
{
     if(data[k] > data[k + 1])
          return false;
}
return true;
```

III.
```
for(int k = 0; k < data.length – 1; k++)
{
     if(data[k] > data[k + 1])
          return false;
     else
          return true;
}
return true;
```

(A) **I only**

(B) II only

(C) III only

(D) I and II only

(E) I and III only

15. Consider the following incomplete method that is intended to return an array that contains the contents of its first array parameter followed by the contents of its second array parameter.

```
public static int[] append(int[] a1, int[] a2)
{
     int[] result = new int[a1.length + a2.length];

     for(int j = 0; j < a1.length; j++)
          result[j] = a1[j];

     for(int k = 0; k < a2.length; k++)
          result[ /* index */ ] = a2[k];

     return result;
}
```

Which of the following expressions can be used to replace `/* index */` so that append will work as intended?

(A) `j`

(B) `k`

(C) `k + a1.length - 1`

(D) **`k + a1.length`**

(E) `k + a1.length + 1`

16. Assume that a and b have been defined and initialized as int values. The expression

```
!(!(a != b ) && (b > 7))
```

is equivalent to which of the following?

(A) `(a != b) || (b > 7)`

(B) **`(a != b) || (b <= 7)`**

(C) `(a == b) || (b <= 7)`

(D) `(a != b) && (b <= 7)`

(E) `(a == b) && (b > 7)`

17. If a, b, and c are integers, which of the following conditions is sufficient to guarantee that the expression

```
a < c || a < b && !(a == c)
```

evaluates to true?

(A) `a == c`

(B) `a == b`

(C) **`a < c`**

(D) `a < b`

(E) `a > b`

18. Consider the following code segment

```
int[] arr = {1, 2, 3, 4, 5, 6, 7};

for(int k = 3; k < arr.length - 1; k++)
          arr[k] = arr[k + 1];
}
```

Which of the following represents the contents of `arr` as a result of executing the code segment?

(A) `{1, 2, 3, 4, 5, 6, 7}`

(B) `{1, 2, 3, 5, 6, 7}`

(C) **`{1, 2, 3, 5, 6, 7, 7}`**

(D) `{1, 2, 3, 5, 6, 7, 8}`

(E) `{2, 3, 4, 5, 6, 7, 7}`

19. ~~Consider the following method that will access a square matrix `mat`:~~

```
/**
 * Precondition: mat is initialized and is a square matrix
*/
public static void printSomething(int [][] mat)
{
   for(int r = 0; r < mat.length; r++)
   {
         for(int c = 0; c <= r; c++)
               System.out.print(mat[r][c] + " ");

         System.out.println();
   }
}
```

~~Suppose mat is originally~~

```
0 1 2 3
4 5 6 7
3 2 1 0
7 6 5 4
```

(A)

```
0
4 5
3 2 1
7 6 5 4
```

(B)

```
0 1 2 3
4 5 6
3 2
7
```

(C)

```
0 1 2 3
4 5 6 7
3 2 1 0
7 6 5 4
```

(D)

```
0
4
3
7
```

(E)

```
3
7
0
4
```

20. An algorithm for finding the average of *N* numbers is

```
average = sum/N
```

Where *N* and sum are both integers. In a program using this algorithm, a programmer forgot to include a test that would check for *N* equal to zero. If *N* is zero, when will the error be detected?

(A) When an incorrect result is output

(B) At compile time

(C) **During run time**

(D) At edit time

(E) As soon as the value of *N* is entered

21. Refer to the `incrementWage` method below.

```
public class Worker{

   private String name;
   private double hourlyWage;
   private boolean isUnionMember

   public Worker()
   { /* implementation no shown */}

   public Worker(String aName, double anHourlyWage, boolean union)
   { /* implementation no show n*/ }

   //Accessors getName, getHourlyWage, getUnionStatus are not shown

   /**Permanently increase hourly wage by amt.
   * @param amt the amount of wage increase
   * /
   public void incrementWage(double amt)
   {  / *implementation of incrementWage */ }

   /**Switch value of isUnionMemeber from true to false
   *  and vice versa
   * /
   public void changeUnionStatus()
   { / *implementation of changeUnionStatus */ }

}
```

Which of the following is correct?

(A) **`hourlyWage += amt;`**

(B) `getHourlyWage() += amt;`

(C) `return hourlyWage + amt;`

(D) `return getHourlyWage() + amt;`

(E) `hourlyWage = amt;`

22. Airmail Express charges for shipping small packages by integer values of weight. The charges for a weight w in pounds are as follows:

`0 < w <= 2`	`$4.00`
`2 < w <= 5`	`$8.00`
`5 < w <= 20`	`$15.00`

The company does not accept packages that weigh more than 20 pounds. Which of the following represents the best set of data (weights) to test a program that calculates charges?

(A) -1, 1, 2, 3, 5, 16, 20

(B) **-1, 0, 1, 2, 3, 5, 16, 20, 22**

(C) 0, 2, 5, 20

(D) 1, 4, 16

(E) All integers from -1 through 22

23. This question is based on the following declarations:

```
String strA = "CARROT", strB = "Carrot", strC = "car";
```

Given that all uppercase letters precede all lowercase letters when considering alphabetical order, which is true?

(A) `strA.compareTo(strB) < 0 && strB.compareTo(strC) > 0`

(B) `!(strA.compareTo(strB) == 0) && strB.compareTo(strA) < 0`

(C) **`strB.compareTo(strC) < 0 && strB.compareTo(strA) > 0`**

(D) `strC.compareTo(strB) < 0 && strB.compareTo(strA) < 0`

(E) `!(strA.compareTo(strB) == 0) && strC.compareTo(strB) < 0`

24. Consider the following code segment

```
int num = 123456789;

while(num > 0){

        int temp = num%10;
        num /= 100;
        System.out.print(temp);
}
```

What will be printed as a result of executing the code segment?

(A) 987654321

(B) 9876543210

(C) **97531**

(D) 75310

(E) 8642

25. Assume that the array `arr` has been defined and initialized as follows,

```
int[] arr = /* initial values for the array */;
```

Which of the following will correctly print all of the odd integers contained in `arr` but none of the even integers contained in `arr`?

(A)
```
for(int x : arr)
        if(x%2 != 0)
             System.out.println(x)
```

(B)
```
for(int k = 1; k < arr.length; k++)
        if(arr[k]%2 != 0)
             System.out.println(arr[k]);
```

(C)
```
for(int x : arr)
        if(x%2!=0)
             System.out.println(arr[x]);
```

(D)
```
for(int k = 0; k < arr.length; k++)
        if(arr[k]%2 != 0)
             System.out.println(k);
```

(E)
```
for(int x : arr)
        if(arr[x]%2 != 0)
             System.out.println(arr[x]);
```

Question 26. Refer to the following code block.

```
public class AutoPart
{
      private String description;
      private int partNum;
      private double price;

      public AutoPart(String desc, int pNum, double aPrice)
      { /* implementation not shown */}

      public String getDescription()
      { return description; }

      public int getPartNum()
      { return partNum; }

      public double getPrice()
      { return price; }

      //Other methods are not shown
      //There is no compareTo method.
}
```

26. Consider the following method

```
/**  Precondition: st1 and st2 are distinct String objects
  *  @return smaller of st1 and st2
  */
public static String min(String st1, String st2)
{
   if(st1.compareTo(st2) < 0)
         return st1;
   else
         return st2;
}
```

A method in the same class has these declarations:

```
AutoPart p1 = new AutoPart(<suitable values>);
AutoPart p2 = new AutoPart(<suitable values>);
```

Which of the following statements will cause an error?

I. `System.out.println(min(p1.getDescription(), p2.getDescription()));`

II.
```
System.out.println(min(((String) p1).getDescription(), ((String)
p2).getDescription()));
```

III. `System.out.println(min(p1, p2));`

(A) I only

(B) II only

(C) III only

(D) **II and III only**

(E) I, II, and III

27. A programmer has a file of names. She is designing a program that sends junk mail letters to everyone on the list. To make the letters sound personal and friendly, she will extract each person's first name from the name string. She plans to create a parallel file of first names only. For example:

fullName	**firstName**
Ms. Anjali DeSouza	Anjali
Dr. John Roufaiel	John
Mrs. Mathilda Concia	Mathilda

Here is a method intended to extract the first name from a full name string,

```
/**Precondition
 * - fullName starts with a title followed by a period
 * - A single space separates the title, first name, and last name
 * @param fullName a string containing a title, period, blank,
 * and last name
 * @return the first name only in fullName
 */
public static String getFirstName(String fullName)
{
   final String BLANK = " ";
   String temp, firstName;

   /* code to extract first name */
   return firstName;
}
```

Which represents correct `/* code to extract first name */`

I.

```
   int k = fullName.indexOf(BLANK);
   temp = fullName.substring(k + 1);
   k = temp.indexOf(BLANK);
   firstName = temp.substring(0, k);
```

II.

```
   int k = fullName.indexOf(BLANK);
   firstName = fullName.substring(k + 1)
   k = firstName.indexOf(BLANK);
   firstName = firstName.substring(0, k);
```

III.

```
   int firstBlank = fullName.indexOf(BLANK);
   int secondBlank = fullName.indexOf(BLANK);
   k = firstName.indexOf(BLANK);
   firstName = fullName.substring(firstBlank + 1, secondBlank + 1);
```

(A) I only

(B) II only

(C) III only

(D) **I and II only**

(E) I, II, and III

28. Refer to the `nextIntInRange` method below:

```
/** @return a random integer in the range low to high, inclusive */
public int nextintInRange(int low, int high)
{
       return /* expression */
}
```

Which `/* expression */` will always return a value that satisfies the post-condition?

(A) **`(int) (Math.random() * (high - low + 1)) + low;`**

(B) `(int) (Math.random() * (high - low)) + low;`

(C) `(int) (Math.random() * (high + low)) + low;`

(D) `(int) (Math.random() * (high + low - 1)) + low;`

(E) `(int) (Math.random() * high) + low;`

Question 29 and 30. Refer to the following code block.

```
public static int mystery(int n)
{
     int x = 1;
     int y = 1;

     //Point A
     while(n>2)
     {
          x = x + y;

          //Point B
          y = x - y;
          n--
     }

     //Point C
     return x;
}
```

29. What value is returned as a result of the call `mystery(6)`?

(A) 1

(B) 5

(C) 6

(D) **8**

(E) 13

30. Which of the following is true of method `mystery`?

(A) `x` will sometimes be `1` at `//Point B`

(B) `x` will never be `1` at `//Point C`

(C) `n` will never be greater than `2` at `//Point A`

(D) `n` will sometimes be greater `2` at `//Point C`

(E) **`n` will always be greater than `2` at `//Point B`**

31. Consider the following code segment.

```
for(int k = 1; k <= 100; k++)
        if((k%4) == 0)
                  System.out.println(k)
```

Which of the following code segments will produce the same output as the code segment above?

(A)
```
for(int k = 1; k <= 25; k++)
      System.out.println(k);
```

(B)
```
for(int k = 1; k <= 100; k = k + 4)
      System.out.println(k);
```

(C)
```
for(int k = 1; k <=100; k++)
      System.out.println(k);
```

(D)
```
for(int k = 4; k <= 25; k = 4*k)
      System.out.println(k);
```

(E)
```
for(int k = 4; k <= 100; k = k + 4)
      System.out.println(k);
```

32. Consider the following method.

```
public static String scramble(String word, int howFar)
{
        return word.substring(howFar + 1, word.length()) +
               word.substring(0, howFar);
}
```

What value is returned as a result of the call `scramble("compiler", 3)`?

(A) compiler

(B) pilercom

(C) **ilercom**

(D) ilercomp

(E) No value is returned because an `IndexOutOfboundsException` is thrown.

33. Consider the following method.

```
public void mystery(int[] data)
{
        for(int k = 0; k < data.length – 1; k++)
               data[k + 1] = data[k] + data[k + 1];
}
```

The following code segment appears in another method in the same class.

```
int[] values = {5, 2, 1, 3, 8};
mystery(values);
for(int v : values)
        System.out.print(v + " ");
System.out.println();
```

What is printed as a result of executing the code segment?

(A) `5 2 1 3 8`

(B) `5 7 3 4 11`

(C) **`5 7 8 11 19`**

(D) `7 3 4 11 8`

(E) Nothing is printed because an `ArrayIndexOutOfBoundsException` is thrown during the execution of method `mystery`

34. Consider the following method

```
public int compute(int n, int k)
{
        int answer = 1;
        for(int i = 1; i <= k; i++)
                answer *= n;
        return answer;
 }
```

Which of the following represents the value returned as a result of the call `computer(n, k)`?

(A) `n*k`

(B) `n!`

(C) n^k

(D) 2^k

(E) k^n

35. Consider the following code segment

```
int sum = 0;
int k = 1;
while(sum < 12 || k < 4)
        sum += k;

 System.out.println(sum);
```

What is printed as a result of executing the code segment?

(A) 6

(B) 10

(C) 12

(D) 15

(E) **Nothing is printed due to an infinite loop**

36. Consider the following code segment

```
int num = 2574;
int result = 0;

while(num > 0)
{
        result = result * 10 + num % 10;
        n /= 10;
 }

 System.out.println(result);
```

What is printed as a result of executing the code segment?

(A) 2

(B) 4

(C) 18

(D) 2574

(E) **4752**

37. Consider the following method

```
public void test(int x)
{
        int y;

        if(x % 2 == 0)
              y = 3;
        else if(x > 9)
              y = 5;
        else
              y = 1;

        System.out.println("y = " + y);
}
```

Which of the following data sets would test each possible output for the method?

(A) 8, 9, 12

(B) 7, 9, 11

(C) **8, 9, 11**

(D) 8, 11, 13

(E) 7, 9, 10

38. Consider the following code segment.

```
int x = 1;
while( /* missing code */ ){
        System.out.print(x + " ");
        x = x + 2;
}
```

Consider the following possible replacements for `/* missing code */`.

I. x < 6

II. x != 6

III. x < 7

Which of the proposed replacements for `/* missing code */` will cause the code segment to print only the values 1 3 5?

(A) I only

(B) II only

(C) I and II only

(D) **I and III only**

(E) I, II, and III

39. Assume that `x` and `y` have been declared and initialized with `int` values. Consider the following Java expression.

```
(y > 10000) || (x > 1000 && x < 1500)
```

Which of the following is equivalent to the expression given above?

(A) `( y > 10000 || x > 1000 ) && ( y > 10000 || x < 1500 )`

(B) `( y > 10000 || x > 1000) || ( y > 10000 || x < 1500 )`

(C) `( y > 10000) && ( x > 1000 || x < 1500 )`

(D) `( y > 10000 && x > 1000) || (y > 10000 && x < 1500 )`

(E) `( y > 10000 && x > 1000) && ( y > 10000 && x < 1500 )`

40. Consider the following incomplete method, which is intended to return the number of integers that evenly divide the integer `inputVal`. Assume that `inputVal` is greater than 0.

```
public static int numDivisors(int inputVal)
{
      int count = 0;

      for (int k = 1; k <= inputVal; k++)
      {
            if ( /* condition */ )
            {
                  count++;
            }
      }
return count;
}
```

Which of the following can be used to replace /* condition */ so that numDivisors will work as intended?

(A) **inputVal % k == 0**

(B) k % inputVal == 0

(C) inputVal % k != 0

(D) inputVal / k == 0

(E) k / inputVal > 0