McGill

Winter 2019
Midterm Examination 1

**INTRODUCTION TO COMPUTER SCIENCE**

**COMP 250**

# Version A

**Thursday February 7th 18:30-20:00**

Examiner: Prof. Martin Robillard Examiner: Dr. Giulia Alberini

Signature: _____ Signature: _____

| Student Name: | | McGill ID: | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|

INSTRUCTIONS:

- This is a **CLOSED BOOK** examination
- Answer **ON THE SCORE SHEET**
- **Ensure that the version number on your score sheet matches the version number of this exam questionnaire.**

**1. Explain what happens when the following statement is compiled and, if applicable, executed.**

```
String result = 8 + "$";
```

A. The Java compiler cannot add a number with a string, and indicates a compilation error
B. When the code executes, the Java environment tries to convert "$" to a number and fails, throwing an exception
C. The literal 8 is converted to the string "8" and concatenated with the string "$" (ANSWER)
D. The Unicode character value of "$" is added to 8 and the resulting character (which is "+") converted to a string.
E. None of the above.

**2. How can you obtain the string "Courses" from the string stored in variable `input`. Remember that in the Java's `substring(int b, int e)` method, `b` is the index of the first character to include in the substring and `e` is 1 more than the index of the last character to include in the substring.**

```
String input = "myCourses";
```

A. `String result = input.substring(2,9);` (ANSWER)
B. `String result = input; result.substring(2,9);`
C. `input.substring(2,9); result = input;`
D. `input.substring(2,9); result = new String(input);`
E. None of the above

**3. Recall the List-Intersection problem from class and the first solution implemented using two nested loops. Which of the following is the worst case scenario for the nested-loops algorithm? Consider that "worst case" means the highest number of comparison needed for the algorithm to terminate.**

A. The two lists are identical.
B. One list is double the size than the other.
C. The two lists have no elements in common. (ANSWER)
D. The two lists contain the same elements, but in the reverse order.
E. None of the above.

**4. What is the character encoding problem?**

A. Hardware problems cause characters to get corrupted when read from a file
B. Some text files are encoded by suspicious characters
C. It is impossible to encode certain characters in files
D. The convention used to encode characters can be ambiguous (ANSWER)
E. In some languages characters must be encoded from right-to-left

**5. Which regular expression matches all strings that start with "http://" or "https://" and only those strings? (as usual, do not include the quotes)**

A. https*://
B. https?://
C. ^https*://
D. ^https?:// (ANSWER)
E. None of the above

**6. What is the purpose of writing assert predicates that constrain the values that function parameters can take?**

A. To clarify what values can be passed into the function or method
B. To make it easier to find bugs
C. To reduce the amount of error-checking logic we need to write inside the function
D. All of the above (ANSWER)
E. None of the above

**7. What is $(1\ 1101\ 1101)_2$ in base 5?**

A. 2043
B. 1042
C. 3402 (ANSWER)
D. 5302
E. 4032

**8. What is not an essential component of a unit test?**

A. A unit under test
B. A unit testing library (ANSWER)
C. Unit data
D. The oracle
E. Assertion code

**9. Which of the following is a valid JUnit test?**

A.
```
public void testMax()
{
    @Test
    assertEquals(6, Math.max(5, 6));
}
```

B.
```
@Test
public void testMax()
{
    assertEquals(6, Math.max(5, 6));
}
```

C.
```
@Test(6)
public void testMax()
{
    Math.max(5, 6);
}
```

D.
```
public void testMax()
{
    System.out.println(Math.max(5, 6) == 6);
}
```

E.
```
public void testMax()
{
    Math.max(5, 6) == @Test(6);
}
```

**10. Consider the following recursive method for checking if the input integer x is even.**

```
public static boolean isEven(int x) {
      int y = Math.abs(x);
      if(y==0)
            return true;
      else
            return ____;
}
```

**Which of the following is the correct return statement?**

A. `isEven(1)`
B. `! isEven(x-1)`
C. `false || isEven(y-1)`
D. `false`
E. `! isEven(y-1)` (ANSWER)

**11. Consider the following array-backed implementation of a list abstract data type. What code is necessary to correctly implement the operation to remove the element at position `position`? Choose the code needed at location X and Y, respectively.**

```
public class ArrayList
{
    public int length = 0;
    public int[] elements = new int[100];

    public static void remove(ArrayList list, int position)
    {
        assert list != null && position < list.length;

        for( int i = position; i < list.length-1; i++  )
        {
            // X
        }
        // Y
    }
```

A. X: Nothing; Y: `list.length--;`
B. X: `list.elements[i-1] = list.elements[i];` Y: Nothing
C. X: `list.elements[i] = list.elements[i+1];` Y: Nothing
D. X: `list.elements[i-1] = list.elements[i];` Y: `list.length--;`
E. X: `list.elements[i] = list.elements[i+1];` Y: `list.length--;` (ANSWER)

**12. How do you remove a node from a doubly-linked list? Assume that `previous` refers to the node in the list before the node we wish to remove, and that each node has a `next` and `previous` field.**

A.
```
previous.next.next.previous = previous;
previous.next = previous.next.next;
```
(ANSWER)

B.
```
previous.next = previous.next.next;
previous.next.next.previous = previous;
```

C.
```
previous.next.previous = previous;
previous.next = previous.next.next.next;
```

D.
```
previous.next = previous.next.next.next;
previous.next.previous = previous;
```

E. None of the above

**13. Which of the following is NOT an example of an algorithm?**

A. A recipe for baking chocolate chip cookies.
B. The list of instructions to follow in order to install Eclipse on your computer.
C. A program that calculates and returns a list of every prime number. (ANSWER)
D. A program that takes in an integer as input and simply outputs the same number.
E. None of the above.

**14. Consider the following algorithm for computing the Fibonacci numbers.**

```
INPUT: An integer n.
OUTPUT: The n-th Fibonacci number.
ALGORITHM:
F[0] = 0
F[1] = 1
For i=2 to n do
      F[i] = F[i-1] + F[i-2]
Return F[n]
```

**Assume that both the assignment and the addition operations take 1 unit of time and ignore other computation cost (for, return, etc.).**

**How many units of time does it take to compute $F_0$?**

A. 1
B. 2 (ANSWER)
C. 4
D. 5
E. 6

**15. Consider the following snippet of code.**

```
String s = "5";
double y = 3;
s = s + y/2;
int z = Integer.parseInt(s);
System.out.println(z);
```

**What is the outcome?**

A. Prints "53.0/2"
B. Prints "25"
C. Prints "26"
D. Compile-time error
E. Run-time error (ANSWER)

**16. Which regular expression matches four or five-digit palindromes? Palindromes are sequences that are read the same way forward and backward, for example 1001 or 12321**

A. \d{5}
B. \d\d\d?\d\d
C. \d\d\D\d\d
D. (\d\d)?\1
E. (\d)(\d)\d?\2\1 (ANSWER)

**17. Consider the following snippet of code.**

```
double[] a = {1.5,3.2,5.8};
int[] b = (int) a;
for (int i=0; i<a.length; i++)
      a[i]++;
System.out.println(Arrays.toString(b));
```

**What is the outcome?**

A. Prints "[1, 3, 5]"
B. Prints "[2, 4, 6]"
C. Prints "[1, 3, 6]"
D. Compile-time error (ANSWER)
E. Run-time error

**18. Consider the following snippet of code.**

```
public static void main(String[] args) {
      String s = "testing ";
      for(int i=0; i<5; i++)
            count(s);
      System.out.println(s.length());
}

public static String count(String s) {
      return (s + 123);
}
```

**What is the outcome?**

A. Prints 8 (ANSWER)
B. Prints 11
C. Prints 23
D. Compile-time error
E. Run-time error

**19. What is the best way to determine if two variables `a` and `b` of type `String` refer to objects that represent the same sequence of characters? Assume neither value is `null`.**

A. To check if `a.length == b.length`, then check that each character matches using `a.charAt[i]` in a for loop.
B. To use a recursive algorithm that uses `a.substring` to compare one half of the string at the time
C. With the statement `a == b;`
D. With the statement `a.toCharArray() == b.toCharArray();`
E. With the statement `a.equals(b);` (ANSWER)

**20. Below is the CORRECT version of the gcd algorithm with no logical errors. Say you run this function with inputs n1 = 9, n2 = 26, and you set a breakpoint at line 4. What is the value stored in variable n2 when the program reaches this breakpoint (before it executes the line).**

```
1    public static void gcd(int n1, int n2) {
2       while(n1 != n2) {
3               if (n1 > n2) {
4                       n1 -= n2;
5               } else {
6                       n2 -= n1;
7               }
8       }
9       System.out.println(n1);
10   }
```

A. 9
B. 26
C. 8 (ANSWER)
D. 17
E. None of the above

**21. Consider the following snippet of code.**

```
public static void main(String[] args) {
      int[] one = {3,5,7};
      int[] two = {4,6,8};
      int[][] a = {one, two};
      mistery(a, one, two);
      System.out.println(Arrays.deepToString(a));
}

public static void mistery(int[][] b, int[] one, int[] two) {
        int[][] a = new int[2][3];
        b[1] = a[0];
        a[0] = two;
        a[1] = one;
        for(int i=0; i<3; i++) {
            one[i]++;
            two[i] *= 2;
        }
}
```

**What does it print? `Arrays.deepToString()` returns a string representation of the objects in the array.**

A. `[[4, 6, 8], [3, 5, 7]]`
B. `[[4, 6, 8], [0, 0, 0]]` (ANSWER)
C. `[[8, 12, 16], [4, 6, 8]]`
D. `[[3, 5, 7], [4, 6, 8]]`
E. `[[0, 0, 0], [8, 12, 16]]`

**22. How many bits do we need to represent $(513)_{10}$ in base 2?**

A. 7
B. 8
C. 9
D. 10 (ANSWER)
E. 11

**23. In computing, what does the acronym ADT commonly refer to?**

A. Analog-Digital Terminal
B. Advanced Data Transaction
C. Automatic Decimal Translation
D. Abstract Data Type (ANSWER)
E. Addition-Division Technique

**24. What is the best definition for an object, in the context of object-oriented programming?**

A. A graph of elements related to each other with references.
B. A reference to a data element.
C. A group of piece of data in memory that belongs together. (ANSWER)
D. A list of fields and methods.
E. An element in an array.

**25. Consider the following class declaration.**

```
public class ArrayList
{
  private int length = 0;
  private int[] elements = new int[100];

  public static int get(ArrayList list, int position)
  { return list.elements[position]; }
}
```

**What is the correct way to implement get as an instance method?**

A. public int get(int position)
{ return this.elements[position]; } (ANSWER)

B. public static int get(int position)
{ return this.elements[position]; }

C. public static int get(ArrayList list, int position)
{ return this.elements[list.position]; }

D. public int get(ArrayList list, int position)
{ return this.elements[list.position]; }

E. None of the above.

**26. Which of the following Java statements will *not* compile. Assume class ArrayList is a subtype of List.**

A. `ArrayList list2 = new List();` (ANSWER)
B. `List list = new ArrayList();`
C. `Integer five = 5;`
D. `int five2 = new Integer(five);`
E. None: they will all compile.

**27. What is the effect of the implements keyword in the following declaration?**

```
public class ClassA implements interfaceB
```

A. It requires ClassA to provide an implementation for all methods declared in interface.
B. It makes ClassA a subtype of interface.
C. It makes it possible to store references to objects of ClassA in variables of type interface.
D. All of the above (ANSWER)
E. None of the above

**28. What is the effect of declaring a field private in Java, as opposed to public?**

A. It is not possible to modify the value of a private field.
B. Private fields cannot be accessed by code in classes other than the class declaring the fields. (ANSWER)
C. Private fields cannot be accessed by static methods.
D. Private fields cannot be accessed by instance methods.
E. None of the above.

**29. Consider the following snippet of code.**

```
public static void main(String[] args) {
      String funnyQuote = "It's not a bug, " +
                          "it's an undocumented feature.";
      char[] some = new char[5];
      for(int i=0; i<5; i++) {
            some[i] = funnyQuote.charAt(i+11);
      }
      String first = new String(some);
      int n = funnyQuote.length();
      String second = funnyQuote.substring(n -8, n);
      System.out.println(first + second);
}
```

**What is the outcome?**

A. Prints "It's a bug, "
B. Prints "It's feature."
C. Prints "bug, feature." (ANSWER)
D. Compile-time error
E. Run-time error

**30. You are asked to design a *singly*-linked list that can add an element to the *end* of the list in constant time (that is, in a fixed amount of time not proportional to the length of the list). What is your strategy?**

A. This is impossible: singly-linked lists require a number of operations proportional to the length of the list to add an element at the end of the list.
B. Make the list circular, so that the next node of the last node is the first node in the list.
C. Use a dummy node, so that the head of the list refers to a node that is not an element of the list.
D. Include an instance variable that refers to the last node in the list as part of the list data structure. (ANSWER)
E. None of the above.