
Instructions
- Answer all questions **on this paper in the spaces provided.**
- No aids, including calculators, are permitted.
- The total number of marks is **72**

Name: _____

Student Number: _____

Section: _____

PART 1 – Short Answer – 32 marks

[2] 1. What would be the output of the following code?

```
int [] arrayA = {1,3};
int [] arrayB = {1,2};
arrayB[1]++;
System.out.println(arrayA == arrayB);
```

[2] 2. A friend of yours likes robots, and built a `Robot` class as part of a project. Unfortunately, it is incomplete. Help your friend out by completing the `clone()` method.

```
class Robot {
    private int serialNum;
    private int servoCount;

    public Robot(int serialNum, int servoCount) {
        this.serialNum = serialNum;
        this.servoCount = servoCount;
    }

    public Robot clone() {

        _____

    }
}
```

[2] 3. Using the syntax for array literals (such as `{1,3,4}`), and not the word `new`, write the declaration statement that would give exactly the same results as `int[][] x = new int[3][2];`

_____

[2] 4. Using the syntax for array literals (such as `{1,3,4}`), write the declaration statement that would give a 2-dimensional array of characters equivalent to `String list = {"java","test"};`

_____

[2] 5. In the `BufferedReader` class, the `String readLine()` method indicates that it has reached the end of the file by returning _____, but the `int read()` method indicates that it has reached the end of the file by returning _____.

[2] 6. Given the declaration `double gpa;` write a statement to throw a new `ArithmeticException` if the value of `gpa` is less than 0. The exception's message should be `"GPA cannot be negative"`.

[2] 7. Given the declarations `Object x; int n;` write the code below to match the comment.

```
//If x refers to a String, then set n to the length of the String,
//otherwise set n to 0.
```

continued…

---

[2] 8. Which of the following statements are true? Circle the correct answers.

a) An abstract class can be a subclass of another class. True False

b) A call to a superclass's overridden method must be the first statement in the subclass method. True False

c) It is always necessary to implement the default (no parameters) constructor in a superclass. True False

d) An object of a subclass type can be assigned to a variable of the superclass type. True False

[2] 9. Rewrite the loop below, assuming that `a` has been changed from type `String[]` to type `ArrayList<String>`.

```
for(int i=0; i<a.length-1; i++)
   a[i] = a[i+1];
```

[2] 10. The following `ArrayList` declaration is meant to hold a list of numbers. Unfortunately, the company intern who wrote it has not yet completed COMP 1020, and made a mistake that results in a **compile-time error**. Help the intern out by **underlining** the incorrect line and then giving the correct line in the space below.

```
ArrayList<int> list = new ArrayList<int>();
for(int i = 0; i < 50; i++)
   list.add(i * 2);
```

[2] 11. Given the definitions of the `Node` and `LList` classes below, which define a linked list of `Objects`, complete the `addToFront` method so that it will add a new `Object` to the front of a list.

```
public class Node {
  private Object data;
  private Node link;
  public Node(Object data, Node link){
    this.data = data; this.link = link;
  }
  public Node getLink() {return link;}
  public Object getData() {return data;}
}

public class LList {
  Node top;
  public LList() { top = null; }
  public void addToFront(Object x){

  }
}
```

continued…

[2] 12. The following is a `contains(Object item)` method from the `LList` class defined in the previous question. It should return `true` if and only if `item` is an element in the list. Unfortunately, the method was emailed to you over a poor WiFi connection and now has two lines missing. Complete the method.

```
public boolean contains(Object item) {
    boolean result = false;
    Node curr = top;

    while(curr != null && !result)
    {
        result = _____________________________________________

        _____________________________________________
    }
    return result;
}
```

[2] 13. What output would be produced by the method shown below if called with `m(2)`?

```
public static void m(int n){
  if(n>8)
    System.out.println("big");
  else {
    System.out.println(n);
    m(2*n+1);
  }
}
```

[2] 14. The following is a *recursive* method designed to perform a linear search of a partially-full array. However, your cloud storage has malfunctioned and parts of the method are missing. Complete the method below by filling in the missing base case and the missing recursive case.

```
public static int linearSearch(int[] values, int n, int key) {
   //returns the index where key appears in the first n elements
   //of values, or -1 if it is not found.
   if(n <= 0)
      return -1;
   else {
      if(values[n-1] == key)
         return _____________________________________________
      else
         return _____________________________________________
   }
}
```

[2] 15. Which sorting algorithm is described by each of the following?

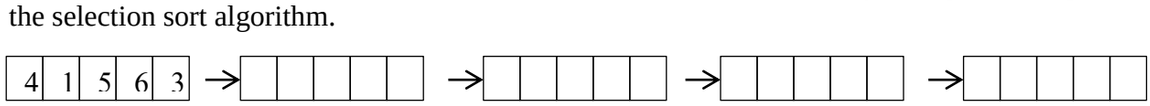
It repeatedly finds the smallest remaining element. _____

It combines two small sorted lists into one larger one. _____

It makes the first k elements sorted, for increasing k. _____

It first splits the list into smaller elements and larger ones. _____

[2] 16. Given the declaration `int[] list = {4,1,5,6,3};` , show the array after performing each step of the selection sort algorithm.



continued…

PART 2 – Programming – 40 marks

[5] 17. Write a class `Movie` with instance variables `name` and `genre` of type `String`, `averageRating` of type `double`, and `numberOfRatings` of type `int`. Define a constructor to initialize `name` and `genre` to two given `String` values (`averageRating` and `numberOfRatings` should initially be 0). Add a `public void addRating(int)` method which will accept a new rating from a movie viewer, and update the rating of the movie accordingly. (To update an average, multiply the current average by the current number of items, add the new item, and divide by the new number of items.) Add a method `public double getRating()` to return the rating of the movie.

continued…

---

[5] 18. Complete the method `mostZero` below, which will return the row number (index) of the row in the two-dimensional array of integers `data` which contains the most zeros. Note that the array `data` may be a ragged array, in which the rows may have different lengths. For example, given the declaration `int[][] test = { {0,3,4,0,8},{0},{3,0,4,0,5,0,6},{1,2,3}};` the call `mostZero(test)` should return 2 since the row with index 2 contains the highest number of zeros (3 of them).

```
public static int mostZero(int[][] data) {
```

continued…

---

[5] 19. As part of your job interview for Maths 'R' Us, you are asked to write a method `int[][] readMatrix(String filename)` which reads integers from the named file, stores them in a rectangular (not ragged) 2D array, and returns that array. The first line of the input file is an integer giving the number of rows in the matrix, and the second line is an integer giving the number of columns. Each remaining line is a *single* integer element of the matrix (row by row). Populate the matrix with these values, and then return the array. You may assume that the number of rows and columns will never be zero. If any I/O errors occur, the method should print `"File read error"`, and any further information contained in the Exception, and return `null`.

```
public static int[][] readMatrix(String filename)
{
```

---

[5] 20. Create an abstract class `Tea`. Create two subclasses `WhiteTea`, and `BlackTea`. The `Tea` class should have a private instance variable `name` of type `String`. All of these classes should have a constructor to initialize `name` to a `String` passed as a parameter. All tea types, other than black tea, have a brewing temperature of 95, whereas black tea has a brewing temperature of 98. Add a method `public int brewTemp()` to the superclass and each subclass (if appropriate) to return the proper brewing temperature of each tea type. (This temperature is *not* stored as an instance variable.)

continued…

---

[5] 21. Write a `checkTens` method which will accept an `ArrayList` of integers (`data`). It should move every multiple of 10 that appears in `data` to the front of the list, unless it's also a multiple of 100, in which case it should be deleted entirely. The multiples of 10 can be placed at the beginning of the list in any order. For example, if `data` contained [35, 30, 99, 200, 67, 20] then `checkTens(data)` should alter it to [20, 30, 35, 99, 67]. (The 20 and 30 could be reversed.) This method does not return anything.

continued…

---

[5] 22. A `Node` class and a partial `LinkedList` class have been provided below. Complete the `printSkip` method which starts by printing the first item in the list (if it exists), and then prints out every (`skip`)th element after that. For example, calling `printSkip(3)` will print the elements at positions 0, 3, 6, 9, and so on, until the list ends. You may assume that `skip` will always be greater than zero.

```
public class Node {
  private int data;
  private Node next;
  public Node(int data, Node next){
    this.data = data; this.next = next;
  }
  public Node getNext() {return next;}
  public int getData() {return data;}
}


public class LinkedList {
    private Node first;

    public LinkedList() {
        first = null;
    }

    public void printSkip(int skip)
    {









    } // printSkip

    // other methods such as add(), etc., not shown
} // LinkedList class
```

continued…

---

[5] 23. Complete the *recursive* method `concatEvenValues`, below, which returns a `String` containing all of the *even* elements of `values`, separated by single blanks. (It's OK to have an extra blank at the end.) Do not do any output. For example, given the declaration `int[] a = {1,2,9,3,4,0};` , the expression `concatEvenValues(a)` should return `"2 4 0 "`.

```
public static String concatEvenValues(int[] values)
{
    return concatEvenValues(values, 0);
}

public static String concatEvenValues(int[] values, int index)
//return a String containing all of the even numbers in values,
//from values[index] to the end of the array.
{
```

continued…

---

[5] 24. Complete the method below to implement a *selection sort* algorithm to sort an `ArrayList` of `Double` values into ascending order.

```
public static void selSort(ArrayList<Double> a){
//Sort the elements of the ArrayList a into ascending order
//using a selection sort algorithm.
```

** THE END **