

1. Section, Date and Time:

CSC 220.02 + .03 - Due **05-16-2020 at 11:55 PM**Full Name in **CAPITAL LETTERS** | **SFSU ID**

--	--

2. Final Exam (1 exam, 0 dropped): 100 points

3. To prepare for this exam, please review all the related materials including WEEK 01-16 packages, slides, mock-up exam(s), reading assignments, in-class practices, sample programs posted in the File Manager, and assignments.
4. You do not need to print this exam. No paper. No handwriting. No scanning. Please type up all your answers in the answer space available in the exam. The provided exam will be in Microsoft Word format. Please submit a single PDF via iLearn.
5. All the rules of an actual exam apply to this exam such as: closed books, closed notes, and no communication with anyone except the course instructor. The course instructor will be available on Zoom (zoom.ucta.net) or via email during the exam time.
6. Please ask all your questions, if any, during the review sessions. Thank you.

HONOR CODE:- Please follow the CS Department's policies: <https://cs.sfsu.edu/student-policies>- Please follow the course's policies: http://csc220.ucta.net/00-README-StudentConduct_AcademicHonesty.pdf

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122

PART A – 50 Points**A.1** - 5 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*How does `compareTo()` work differently when comparing 2 `String` objects versus comparing 2 `enum` objects?

A.2 - 5 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

The SFSU One Stop office asks you to recommend a data structure to manage the line of students they serve. Which one will you use? Please explain why. ☐ **Stack** ☐ **Queue** ☐ **Deque** ☐ **Priority Queue**

A.3 - 5 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

What is the output? And **why**?

```
String a, b;  
String c = new String("Google");  
  
a = "GooglePixel4XL";  
b = "GooglePixel4a";  
System.out.println(a.compareTo(b));
```

OUTPUT:

WHY:

```
a = "AndroidStudioOreo";  
b = "AndroidStudioPie";  
System.out.println(b.compareTo(a));
```

OUTPUT:

WHY:

```
a = "TESLax";  
b = "TESLay";  
System.out.println(b.compareTo(a));
```

OUTPUT:

WHY:

```
a = "Kotlin";  
b = "KotlinX";  
System.out.println(a.compareTo(b));
```

OUTPUT:

WHY:

```
System.out.println(a.compareTo(c));
```

OUTPUT:

WHY:

A.4 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Explain the 3 methods used to make **Deep Copies**.

1.

2.

3.

The SFSU University Police Department has been sending out live updates when there are incidents happening on campus. Would you use **Deep** or **Shallow** copies to implement this communication? Please explain in detail.

A.5 - 5 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

How is **enum** a special type? Please briefly explain 3 characteristics of **enum**.

1.

2.

3.

A.6 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

How does **Insertion Sort** work? What are the steps?

8 2 9 7 6 5 1 3 4

Show the contents of the array above **each time** an **Insertion Sort** changes it while sorting the array into **ascending order**.

[illegible]

A.7 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

You are hired to create a simple Dictionary application. Your dictionary can take a search key from users then returns value(s) associated with the key.

- Please explain **how** you would implement your dictionary.
- Please state clearly **which data structure(s)** you will use and explain your decision.
- Please state clearly **at least 3 methods** you will use and explain your decision.

PART B – 50 Points

B.1 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

What is a **deque**? How does it work? Use an example and diagrams to explain its operations.

Display the **Deque** at the marked lines.

```
Deque d = new LinkedList<>();  
d.addLast("T");  
d.addFirst("e");  
d.add("s");  
d.addFirst("L");  
d.addLast("a");  
d.offer(d.remove(d.contains("T"))); // 1  
d.push(d.remove("S".toLowerCase())); // 2  
d.addLast(d.remove(d.contains("Z"))); // 3  
d.offer(d.remove()); // 4  
d.offerFirst(d.remove(d.element())); // 5
```

FRONT

1.	
2.	
3.	
4.	
5.	

B.2 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

What is a **list**? How does it work? Use an example and diagrams to explain its operations.

Display the List at the marked lines.

```
LinkedList li = new LinkedList();  
for (int i = 9; i >= 2; i--) {  
    li.addLast(i);  
}  
li.remove(li.get(4));  
li.set(6, li.size() % 10);  
li.add(li.size());           // 1  
li.addFirst(li.get(li.size() - 4)); // 2  
li.add(li.indexOf(5));       // 3  
li.addFirst(li.remove(li.indexOf(6))); // 4  
li.addFirst(li.peekLast());  // 5  
System.out.println(li);
```

FRONT

1.	
2.	
3.	
4.	
5.	

B.3 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Write a statement to create a **Google Guava Multimap** instance.

And write code to input 5 entries into this data structure.

And use **keyset()** to display all the contents stored in this data structure.

B.4 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Given a **queue q** of String elements, please write code to output:

- “*Every other.*” if the elements are organized in these patterns:
 - a, OK, b, OK, c, OK, d
 - a, OK, b, OK, c, OK, d, OK
 - OK, a, OK, b, OK, c, OK, d
 - OK, a, OK, b, OK, c, OK, d, OK
- “*NOT every other.*” if other patterns.

B.5 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Given 2 Priority Queues, please write code to detect if one queue is the reverse of the other. Our program should output meaningful message(s).

PART C – 5 Extra Credit Points

Class **Entry** has 1 property: String **finalExam**. Please write the **hashCode** method for this class using **Joshua Bloch's** recommendation.

And write the **equals()** method to compare 2 objects of this class.