

universal cover for all hwks and exams/V2	
Last name, First name	
course (CSCI-NNN)	cs220
Assignment number (hwkN/examN)	hwk3
Date	2021/02/28
your email address	@eagles.bridgewater.edu

Do not do any computation on this page.

If the hwk has Multiple choice questions (MCQ), then mark your solution in this table here.

Suppose you think correct answer is C. You will then attach letter X to your desired answer like CX. If the work is handwritten you can handwrite X, if the work is typed in Word you will type in Word X.

1	A	B	C	D	
2	A	B	C	D	
3	A	B	C	D	
4	A	B	C	D	
5	A	B	C	D	
6	A	B	C	D	
7	A	B	C	D	
8	A	B	C	D	
9	A	B	C	D	
10	A	B	C	D	
11	A	B	C	D	
12	A	B	C	D	
13	A	B	C	D	
14	A	B	C	D	
15	A	B	C	D	

//your typed or handwritten work starts on the next page

Note if I request that your answer should be 4-6 sentences, it is only a guideline. It means if you give me 2 sentences, it may not be enough. If you write half a page it is way too much.

Program1

Write a Java program .

Class name StackChar.

Package: no package.

All classes have class header.

All methods have method comment. For now usually 2-3 sentences are enough. Later may need more.

The StackChar is similar to StackInt we did.

In addition, it will have one other method. This method just helps me to see what you are doing.

void prints()

This method will print all chars that are currently in the stack from bottom of the stack to the top. The data will be printed on a single line with the bottom of the stack on the left.

So if you pushed x then y,

Your prints() method after y will display

>x y

The prints() method will only print the valid data that is in the stack.

Separate by 4 spaces.

In the main you will perform exactly the ops described in that sequence.

```
static public void main(String[] args)
{
    StackChar sc = new StackChar(10);

    //after each stack method push, pop, peek, you will use prints().

    //push 6 a-f letters of alphabet in normal order
    //THEN
    //push the next letter of alphabet
    //pop 1
    //push the next letter of alphabet
    //pop 2
    //push the next letter of alphabet
    //pop 3
    //peek and print what the peek gives.


    //your stack must be able to handle any kind of sequence
    //the grader may use different sequence to test your stack

}
}
```