```
tion 1
  Question#1: What is the output for the following code segment?
    1 \text{ int } n = 1, i = 0;
    2 while(n <= 4){
          switch (n){
              case 1: System.out.print("*");
              case 2: System.out.print("$"); break;
              case 3: System.out.print("@");
              default: System.out.print("%");
          System.out.println();
          if (n >= 3)
   10
   11
              break;
          for(i = 0; i < 3; i++){
   12
              if (i == 1)
   13
   14
                   continue;
              System.out.println(i + "+" + n); }
   15
   16
          n++;
   17 }
   18 System.out.println("n=" + n);
```

The answer at the end

```
Question#3: Write Java statements; limit the number of statements you use.

Assume all variables are already declared and have values unless otherwise specified.

A. Write one Java statement that equivalents to the following formula: x = e^{|2x-4x|}
```

x = Math. exp (Math. abs (2xx-4xx));

```
Question 2

Question#2: I. Does this loop cause an error? What kind? Why?

for(int i=0 , j=1 ; i<2 ; i/=j , j--)

System.out.print("Smile");
```

yes, a runtime error because of the division by Zero in the i/=j at third iteration of the loop

## Question 3 Question#2: II. What is the output of the following Java statement? System.out.print(Math.floor(-2.6));

= -3.0

estion 5 1 out of 1 points

Question#3: Write Java statements; limit the number of statements you use.

Assume all variables are already declared and have values unless otherwise specified.

**B.** Using **conditional operator**, write **one** java statement that prints "even" if an integer variable **num** is an even number or "odd" if it is an odd number.

System. out. print (((num%2) = = 0)? "even"; "odd");

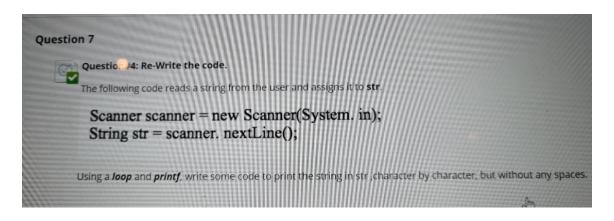
Question#3: Wille Java statements; limit the number of statements you use.

Assume all variables are already declared and have values unless otherwise specified.

**C.** Consider the given declaration of **str\_amount**, and **half\_amount**. Write **one** Java statement to store half the amount given in the **string str\_amount**.

String str\_amount = "15";
double half\_amount;

half\_amount = Double. parseDouble (Str\_amount)/2;



for (int i=0; i \Str.length(); i++)

if (str.charAt(i)!=")

system.out.printf("%c", str.charAt(i));

Question#5: Fill in blanks.
Assume we have the following variable declarations and initializations:
String membershipType = "Student"; char plan = 'A'; double rate = 70.5;
Fill in the below blanks to produce an expected output according the following format:
Student~**~~a**70.50~RS
Note: We used ~ to indicate a SPACE.
System.out.printf("%[A]s**%(B]c**%(C]fRS",MembershipType, [D];rate);

A: -8 B: 3 C: -6.2

D: Character. to Lower Case (Plan);

Q1:

out put: -

$$n = 3$$