

tion 1

Question#1: What is the output for the following code segment?

```
1 int n = 1, i = 0;
2 while(n <= 4){
3     switch (n){
4         case 1: System.out.print("*");
5         case 2: System.out.print("$"); break;
6         case 3: System.out.print("@");
7         default: System.out.print("%");
8     }
9     System.out.println();
10    if (n >= 3)
11        break;
12    for(i = 0; i < 3 ;i++){
13        if (i == 1)
14            continue;
15        System.out.println(i + "+" + n); }
16    n++;
17 }
18 System.out.println("n=" + n);
```

The answer at the end

stion 4

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: $x = e^{|2x-4x|}$

$x = \text{Math.exp}(\text{Math.abs}(2 \times x - 4 \times x));$

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

Question 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: Write 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** into **half_amount**.

```
String str_amount = "15";  
double half_amount;
```

```
half_amount = Double.parseDouble(str_amount)/2 ;
```


Question 7



Question#4: Re-Write the code.

The following code reads a string from the user and assigns it to **str**.

```
Scanner scanner = new Scanner(System.in);  
String str = scanner.nextLine();
```

Using a **loop** and **printf**, write some code to print the string in **str** character by character, but without any spaces.

```
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 to 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.toLowerCase(plan);

Q1:

output:-

*\$

0+1

2+1

\$

0+2

2+2

@%

n=3