AP Computer Science A Name: Mustafa Sarp Gerzen**Unit 2A Review - Selection (2.1-2.6)**

2.1 Algorithms with Selection and Repetition

Define the following terms.

sequencing Doing steps in a specific order	Selection Making a choice in a program based on a condition
repetition Repeated process until a condition changes it	Algorithm A step by step logic behind how a think will workout

What is the purpose of selection statements when programming?
To let program choose between alternative acts based on the conditions and to control a flow of execution.

2.2 Boolean Expressions

Comparison Operators

==	Check if two are equal	!=	Check if two are not equal
<=	Check if one bigger or equal	<	Check if one bigger
>=	Check if one smaller or equal	>	Check if one smaller

Write conditional statements to mean the following.

num is even Num % 2 ==0;	num is odd Num % 2 ==1;
num is a multiple of x Num %x ==0;	num is between min and max num > min && num < max

50% of the time 20% of the time

2.3 If Statements

Write a program that takes a positive integer as input. The program must determine if the number is even or odd. If the number is even, the program should print a message stating that it is even. Otherwise (if it's odd), it should print a message stating that it is odd.	`Scanner in = new Scanner(System.in); int num = in.nextInt(); if (num%2 ==0){ System.out.println(“even”); } else{ System.out.println(“odd”); } }`

A student takes a single test. The passing score is 70. Write a program that takes the student's numerical test score as input. The program must check if the score is 70 or higher. If the score is 70 or greater, the program should output "PASS." Otherwise (if the score is below 70), the program should output "FAIL."	`Scanner in = new Scanner(System.in); int score = in.nextInt();`
	
	`if (score >= 70) {`
	`  System.out.println("PASS");`
	`} else {`
	`  System.out.println("FAIL");`
	`}`

2.4 Nested If Statements

Write a program that takes a student's numerical score as input. The program should then determine and print the corresponding letter grade based on the following scale:
 	
● A score of 90 or above is an 'A'.
● A score between 80 and 89 is a 'B'.
● A score between 70 and 79 is a 'C'.
● A score between 60 and 69 is a 'D'.
● Any score below 60 is an 'F'.
 	
The program must handle all possible score inputs and provide the correct grade.
	`Scanner in = new Scanner(System.in); int grade = in.nextInt();`
	
	`if (grade >= 90) {`
	`  System.out.println("A");`
	`} else if (grade >= 80) {`
	`  System.out.println("B");`
	`} else if (grade >= 70) {`
	`  System.out.println("C");`
	`} else if (grade >= 60) {`
	`  System.out.println("D");`
	`} else {`
	`  System.out.println("F");`
	`}`

AP Computer Science A Name: _____ **Unit 2A Review - Selection (2.1-2.6)**

A movie theater offers discounts based on a customer's age. The standard ticket price is $12. Write a program that takes a customer's age as input.

- If the customer is 12 years old or younger, they get a discount, and the ticket price is $7.
- If they are 65 years old or older, they also get a discount, and the ticket price is $9.
- For all other ages, they should pay the full price.

The program should calculate and display the final ticket price.

```
Scanner in = new
Scanner(System.in); int age =
in.nextInt();

int price;
if (age <= 12) {
    price = 7;
} else if (age >= 65) {
    price = 9;
} else {
    price = 12;
}
System.out.println(price);
```

2.5 Compound Boolean Expressions

Logical	Java	When true?	When false?
and	&&	Both are true	One is false
or	||	At least one is true	Both are false
not	!	Operator is false	Operator is true

Write a program that takes a vehicle's fuel efficiency in Miles Per Gallon (MPG) as input. The program needs to categorize and print the vehicle's rating based on the following criteria:
 	
● If the MPG is 30 or higher, the rating is "Excellent Fuel Economy."
● If the MPG is between 20 and 29 (inclusive), the rating is "Average Fuel Economy."
● Otherwise (if the MPG is below 20), the rating is "Poor Fuel Economy."
	`Scanner in = new Scanner(System.in);`
	`int mpg = in.nextInt();`
	
	`if(mpg>=30){`
	`System.out.println(“Excellent Fuel Economy”);`
	`}`
	`Else if(mpg >=20){`
	`System.out.println(“Average Fuel economy”);`
	
	`Else{`
	`System.out.println(“Poor Fuel Economy”);`
	`}`

AP Computer Science A Name: _____ **Unit 2A Review - Selection (2.1-2.6)**

The program should print the highest level of recognition the student qualifies for based on the following rules:
 	
● Scholar Award: The student qualifies if their GPA is 3.8 or higher AND their Attendance is 95% or higher.
● Dean's List: The student qualifies if they did not get the Elite Scholar Award, but their GPA is 3.0 or higher AND their Attendance is 90% or higher.
● Warning: The student is flagged for a warning if they did not meet either of the above criteria, but their GPA is 2.0 or lower OR their Attendance is 80% or lower.
● Good Standing: If the student meets none of the above specific conditions, they receive "Good Standing."
 	
The program must only output one final status.
	`Scanner in = new Scanner(System.in);`
	`double gpa = in.nextDouble();`
	`int attendance = in.nextInt();`
	
	`if (gpa >= 3.8 && attendance >= 95) {`
	`  System.out.println("Scholar Award");`
	`} else if (gpa >= 3.0 && attendance >= 90) {`
	`  System.out.println("Dean's List");`
	`} else if (gpa <= 2.0 || attendance <= 80) {`
	`  System.out.println("Warning");`
	`} else {`
	`  System.out.println("Good Standing");`
	`}`

2.6 Comparing Boolean Expressions

Use De Morgan's Law to simplify the following:

!(a && b) = !a || !b	!(a || b) = !a && !b

!(x > 5 && y == 6) x<=5 || y !=6	!(x != 7 || y <= 8) x == 7 && y>8

What is the result of the following code segments based on the following variable definitions?

```
String s1 = "Hello"; String s2 = "there";
String s3 = new String("Hello"); String s4 = s1;
```

s1 == s2 false s1 == s3 false s1 == s4 true	s1.equals(s2) false s1.equals(s3) true s1.equals(s4)true
s1.compareTo(s2) < 0 true s1.compareTo(s3) < 0 false s1.compareTo(s4) < 0false	s1.compareTo(s2) > 0 false s1.compareTo(s3) > 0 false s1.compareTo(s4) > 0 false