

JAVA REVIEW SLIDES

DAVE SLEMON, VI00

Q1. Static Class Constants

1. Create a class called, Me.
2. Declare and populate a static class constant called, NUM_FAMILY_MEMBERS and set it to 6;

Me
<u>- NUM_FAMILY_MEMBERS: integer</u>

A1. Static Class Constants

Me
<u>- NUM_FAMILY_MEMBERS: integer</u>

1. Create a class called, Me.
2. Declare and populate a static class constant called, NUM_FAMILY_MEMBERS = 6

```
public class Me
{
    //instance vars & constants
    private static final int NUM_FAMILY_MEMBERS = 6;
}
```

Q2. Rewrite this code so that it works as intended

```
import java.util.Scanner;
public class TestProgram
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter name > ");
        String name = scanner.nextLine();

        if (name == "Dave")
            System.out.println("Hi " + name);
    }
}
```

A2. Rewrite this code so that it works as intended

```
import java.util.Scanner;
public class TestProgram
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter name > ");
        String name = scanner.nextLine();

        if ( name.equalsIgnoreCase("Dave") )
            System.out.println("Hi " + name);
    }
}
```

Q3. Rewrite this code so that it works as intended

Class average is: 7.0

```
import java.util.Scanner;
public class TestProgram
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int quiz_score1 = 5;
        int quiz_score2 = 10;

        double avg = (quiz_score1 + quiz_score2) / 2;
        System.out.println("Class average is: " + avg);
    }
}
```

A3. Rewrite this code so that it works as intended

Class average is: 7.5

```
import java.util.Scanner;
public class TestProgram
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int quiz_score1 = 5;
        int quiz_score2 = 10;

        double avg = ( (double)quiz_score1 + ((double)quiz_score2) / 2;
        System.out.println("Class average is: " + avg);
    }
}
```

CASTING IS REQUIRED
(double)

Q4. Rewrite this code so that it works as intended

Enter your age > 29

Thanks, next year you'll be 291

```
import java.util.Scanner;
public class TestProgram
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter your age > ");
        String age = scanner.nextLine();

        System.out.println("Thanks, next year you'll be " + age+1 );
    }
}
```

A4. Rewrite this code so that it works as intended

Enter your age > 29
Thanks, next year you'll be 30

```
import java.util.Scanner;  
public class TestProgram  
{  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter your age > ");  
        String age = scanner.nextLine();  
        int theAge = Integer.parseInt(age);  
        System.out.printf("Thanks, next year you'll be %d \n", theAge + 1 );  
    }  
}
```

Q5. In a method, how would you declare a constant called,
MAX = 100;

A5. In a method, how would you declare a constant called, MAX = 100;

static final int MAX = 100;

The reason that we have to use both static and final modifiers is that if we declare a variable '**var**' only as static, all the objects of the same class will be able to access this '**var**' and change its values. When we declare a variable only as final, then multiple instances of the same constant value will be created for every different object and that isn't efficient/desirable. When we use both static and final, then, the '**var**' remains static and can be initialized only once, thereby making it a proper constant which has a common memory location for all objects of it's containing class.

Q6. Consider TestProgram.java. Add some code to TestProgram. Declare a variable to Car. The constructor allows for one colour and the make of the car. e.g. Red Toyota. Output the variable object to the terminal window.

A6. Consider TestProgram.java. Add some code to TestProgram. Declare a variable to Car. The constructor allows for one colour and the make of the car. e.g. Red Toyota. Output the variable object to the terminal window.

```
Car t = new Car( Color.RED, "Toyata" );
```

```
System.out.println( t );
```

Q6. Ask a teenager to enter his or her age. Continue to ask the teenager for their age until they enter an age between 13 and 19 inclusive.

You can assume that the user always enters a GOOD integer.

Your program
should output as shown in the sample session below:

```
Enter your age (13-19) > 20  
Enter your age (13-19) > 12  
Enter your age (13-19) > 24  
Enter your age (13-19) > 14  
Thank you, you are 14 years old
```

A6. Ask a teenager to enter his or her age.
Continue to ask the teenager for their age until they enter an age between 13 and 19 inclusive. You can assume that the user always enters a GOOD integer.

Your program should output as shown in the sample session below:

```
Enter your age (13-19) > 20
Enter your age (13-19) > 12
Enter your age (13-19) > 24
Enter your age (13-19) > 14
Thank you, you are 14 years old
```

```
import java.util.Scanner;

public class question9
{
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        int age = 0;
        do{
            System.out.print("Enter your age (13-19) > ");
            age = scanner.nextInt();
            scanner.nextLine(); //clear the line
        } while(age < 13 || age > 19);
        System.out.print("Thank you, you are " + age + "
years old");
    }
}
```