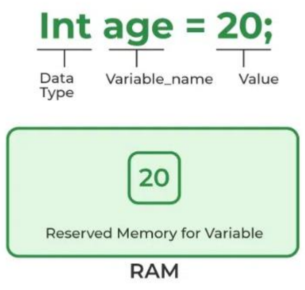
Java Questions

Gwynn & Brian's Questions

1. How do you start a variable?

To declare (create) a variable, you will specify the type, leave at least one space, then the name for the variable and end the line with a semicolon (;). Java uses the keyword int for integer, double for a floating point number (a double precision number), and boolean for a Boolean value (true or false).



2. How do you make a form in Java?

Create a java file containing the main class

```
class Registration {

    public static void main(String[] args)
                                throws Exception
    {
        MyFrame f = new MyFrame();
    }
}
```

create another MyFrame which will contain the form

In MyFrame components like (JLabel, JTextField)

A constructor used to initialize the components

"actionPerformed()" to get the action performed

Compile the file using "javac" command

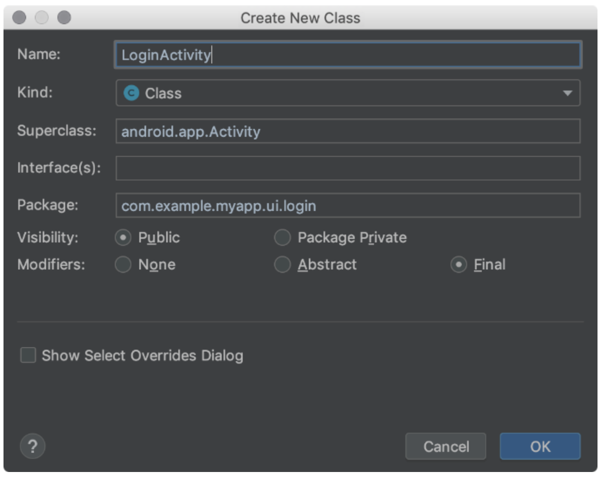
```
javac Registration.java
```

Run program by calling main class

```
java Registration
```

3. How do you set up and start a class?

To create a new Java class or type, follow these steps: In the Project window, right-click a Java file or folder, and select New > Java Class. Alternatively, select a Java file or folder in the Project window, or click in a Java file in the Code Editor. Then select File > New > Java Class.


Create New Class

Name: LoginActivity
Kind: Class
Superclass: android.app.Activity
Interface(s):
Package: com.example.myapp.ui.login
Visibility: [✓ Public] Package Private
Modifiers: None Abstract [✓ Final]
Show Select Overrides Dialog

4. What are some Java restrictions?

Cannot Instantiate Generic Types with Primitive Types

Cannot Create Instances of Type Parameters

Cannot Declare Static Fields Whose Types are Type Parameters

5. What are some of the limitations in making a GUI in Java?

On the other hand, there are some disadvantages to using Java for GUI applications. For one, Java applications tend to be more resource intensive than native applications written in languages like C. Additionally, some features may be more difficult to implement in Java than in a native language like C.

6. Import sometimes has a star...

In Java, the ` ` in imports (e.g., ` import java.util.;`) is a wildcard that imports all the classes from a package instead of specifying each class individually.

Pros of Using ` `:

- Convenience: It saves time when you need multiple classes from a package.

- Cleaner Code: Reduces the number of import statements.

Cons of Using ` `:

- Less Clarity: It's not clear which classes are being used.

- Risk of Conflicts: It can cause issues if classes with the same name exist in different packages.

Using ` ` is handy but can make the code harder to understand.

7. What does the create class mean when trying to fix an error?

When you see a "create class" option while fixing an error, it means you're referencing a class that hasn't been defined yet. For example, if you try to use `Person person = new Person();` but haven't created a `Person` class, the IDE detects this and offers to help by generating a basic class structure for you.

By selecting "create class," the IDE will automatically add the necessary code to create a new class, like this:

```java
public class Person {
   // You can add fields, methods, etc. later
}
```

This feature speeds up development by quickly setting up the class, allowing you to focus on adding the details you need later.

8. Polymorphism...

Polymorphism in Java lets one object take many forms. It allows different classes to respond to the same method in their own way.

Example:

```java
class Animal {
   public void sound() {
      System.out.println("Animal makes a sound");
```

```
    }
}

class Dog extends Animal {
    @Override
    public void sound() {
        System.out.println("Dog barks");
    }
}

class Cat extends Animal {
    @Override
    public void sound() {
        System.out.println("Cat meows");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal myDog = new Dog();
        Animal myCat = new Cat();

        myDog.sound();  // Output: Dog barks
        myCat.sound();  // Output: Cat meows
    }
}
```

Explanation:

Even though `myDog` and `myCat` are both declared as `Animal`, they behave differently because `myDog` is actually a `Dog` and `myCat` is actually a `Cat`. This is polymorphism in action.

9. When to use “void”?

In Java, `void` is used to indicate that a method does not return any value. You use `void` when you want the method to perform an action but don't need to send any data back to the caller.

Example:

```java
public class Main {
    public static void printMessage() {
        System.out.println("Hello, World!");
    }

    public static void main(String[] args) {
        printMessage();  // Calls the method, but it doesn't return anything
    }
}
```

Explanation:

In this example, `printMessage()` is a `void` method because it performs an action (printing a message) but does not return any value.

10. What “&&” means?

In Java, `&&` is the logical AND operator. It is used to combine two boolean expressions and returns `true` only if both expressions are `true`. If either or both expressions are `false`, it returns `false`.

Example:

```java
public class Main {
   public static void main(String[] args) {
      int a = 10;
      int b = 20;

      if (a > 5 && b > 15) {
         System.out.println("Both conditions are true!");
      } else {
         System.out.println("At least one condition is false.");
      }
   }
}
```

Explanation:

In this example, `a > 5 && b > 15` checks if both conditions are true. Since both are true (`a` is greater than 5 and `b` is greater than 15), the message `"Both conditions are true!"` is printed. If either condition had been false, the else block would have executed.

Mpho & Caitlyn's Questions

1. **How do you define all the different java data types?**

   Data type

   Data types are divided in two groups we have Primitive data type and non-primitive data type

   Primitive data type includes **byte, short, int, long, float, double, Boolean and char**

   Non-primitive data types such as **String, Arrays and Classes**

   Numbers

   Primitive number types are divided into two groups which are Integer types and Floating-point types

   Integer types store whole numbers, positive or negative (such as 123 or -456), without decimals. Valid types are byte, short, int and long. Which type you should use, depends on the numeric value.

   Floating point types represents numbers with a fractional part, containing one or more decimals. There are two types: float and double.

   Boolean

   Boolean is a java type that only takes the values **true** or **false**

   Characters

   The **char** data type is used to store a **single** character. The character must be surrounded by single quotes, like 'A' or 'c':

2. **How can you use ActionListener () to get user input in Java?**

   To use ActionListener to get user input in Java, you need to do the following

Implement the ActionListener Interface: Your class needs to implement the ActionListener interface.

Register the Component with the Listener: Add the ActionListener to the component that will trigger the event (e.g., a button or text field).

Override the action Performed Method: Define what should happen when the action is performed.

3. **How can I add 2 labels in Java Swing on separate lines? (Java Swing is what we used for the UI yesterday)**

```
import javax.swing.*;

public class LabelExample {
  public static void main(String[] args) {
    // Create a new frame
    JFrame frame = new JFrame("Label Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 200);

    // Create a panel to hold the labels
    JPanel panel = new JPanel();
    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS)); // Use BoxLayout for vertical alignment

    // Create two labels
    JLabel label1 = new JLabel("Label 1");
    JLabel label2 = new JLabel("Label 2");

    // Add labels to the panel
    panel.add(label1);
    panel.add(label2);

    // Add panel to the frame
    frame.add(panel);

    // Set frame visibility
    frame.setVisible(true);
    }
  }
```

4. **What is the difference between Polymorphism and Inheritance?**

   Polymorphism is that in which we can perform a task in multiple forms or ways. It is applied to the functions or methods. Polymorphism allows the object to decide which form of the function to implement at compile-time as well as run-time.

   Inheritance is one in which a new class is created that inherits the properties of the already exist class. It supports the concept of code reusability and reduces the length of the code in object-oriented programming.

5. **What Java package can be used to make websites and 3D websites?**

   Java 3D API

   The Java 3D API enables the creation of three-dimensional graphics applications and Internet-based 3D applets. It provides high-level constructs for creating and manipulation 3D geometry and building the structures used in rendering that geometry.

1. How to create classes and object in java

```
//Classes are defined with the class keyword

class ClassName {

  // Fields (Instance Variables)

  DataType variableName;


  // Constructors

  public ClassName() {

    // Initialization code

  }


  // Methods are functions defined inside a class. They define behaviors or actions
that the object can perform.
```

```
  ReturnType methodName() {
    // Method body
    return value;
  }
}
```

2. How to link list in Java

The LinkedList stores its items in "containers." The list has a link to the first container and each container has a link to the next container in the list. To add an element to the list, the element is placed into a new container and that container is linked to one of the other containers in the list.

```
// Import the LinkedList class
import java.util.LinkedList;


public class Main {
 public static void main(String[] args) {
  LinkedList<String> cars = new LinkedList<String>();
  cars.add("Volvo");
  cars.add("BMW");
  cars.add("Ford");
  cars.add("Mazda");
  System.out.println(cars);
 }
}
```

Output: [Volvo, BMW, Ford, Mazda]

3. How to create Threads in java

Threads allows a program to operate more efficiently by doing multiple things at the same time. Thet can be used to perform complicated tasks in the background without interrupting the main program.

```
public class Main implements Runnable {
  public static void main(String[] args) {
    Main obj = new Main();
    Thread thread = new Thread(obj);
    thread.start();
    System.out.println("This code is outside of the thread");
  }
  public void run() {
    System.out.println("This code is running in a thread");
  }
}
```

4. How to create java constructs

A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes:

```
// Create a Main class
public class Main {
  int x;  // Create a class attribute

  // Create a class constructor for the Main class
  public Main() {
    x = 5;  // Set the initial value for the class attribute x
  }
  public static void main(String[] args) {
      Main myObj = new Main();  // Create an object of class Main (This will call the
constructor)
      System.out.println(myObj.x); // Print the value of x
  }
}
```

Outputs: 5

5. How to create java Arry List

   int[] myArray = new int[] {1, 2, 3};

   The int[] myArray says that we want to declare a new variable called myArray that has the type of an array of integers.
   new int[] says we want to immediately allocate memory to store data in this array.
   {1,2,3} is the data we want to store in the array.