RAMESH RESHIYA HAS **CREATE** *A FILE*

Awesome! Here's the **full solution** for the **CS–22 Java Prelims Paper** based on the uploaded PDF.

✅ Q.1 (A) One Sentence Answers

Set 1:

1. **IIB** stands for *Instance Initialization Block*
2. **JVM** stands for *Java Virtual Machine*
3. **JRE** stands for *Java Runtime Environment*

Set 2:

1. **OOP** stands for *Object-Oriented Programming*
2. **IDE** stands for *Integrated Development Environment*
3. **NetBeans** is an IDE used for developing Java applications.

✅ Q.1 (B) Short Answers

Option 1: JDK Tools

- **JDK Tools** include: javac (compiler), java (launcher), javadoc (documentation), jdb (debugger), and javap (disassembler).
- **Example**: javac Hello.java compiles Java code into bytecode.

Option 2: Java Operators

- Java has: *Arithmetic*, *Relational*, *Logical*, *Assignment*, *Bitwise*, and *Unary* operators.
- **Example – Arithmetic**: int sum = a + b;

✅ Q.1 (C) Long Answers

Option 1: Constructor

- A **constructor** is a special method called when an object is instantiated.

- **Types**: Default, Parameterized, Copy constructor.
- **Example**:

```
class Example {
  Example() {
    System.out.println("Default Constructor Called");
  }
}
```

Option 2: Inheritance

- **Inheritance** allows a class to inherit properties from another.
- **Types**: Single, Multilevel, Hierarchical, Hybrid.
- **Example**:

```
class A {
  void show() { System.out.println("Class A"); }
}
class B extends A {
  void display() { System.out.println("Class B"); }
}
```

Alternative Option: Command Line Argument

```
class Test {
  public static void main(String[] args) {
    for(String arg : args) {
      System.out.println(arg);
    }
  }
}
```

Static vs Non-static:

- **Static**: belongs to class, shared across all objects.
- **Non-static**: belongs to object.

```
class Example {
 static int x = 10;
 int y = 20;
}
```

✅ Q.2 (A) One Sentence Answers

Set 1:

1. java.lang is the default package imported.
2. Object is the superclass of all classes.
3. False – private members are not inherited.

Set 2:

1. length(), charAt(), toUpperCase()
2. default is the access specifier.
3. True – interface helps achieve multiple inheritance.

✅ Q.2 (B) Short Answers

Option 1: Method Overriding

- When a subclass provides a specific implementation of a method already defined in its superclass.

Option 2: Import Types

- **Normal Import**: import java.util.Scanner;
- **Static Import**: import static java.lang.Math.*;

✅ Q.2 (C) Long Answers

Option 1: Packages

- A **package** is a namespace that organizes classes and interfaces.
- **Built-in**: java.util, java.io, java.lang
- **Example**: import java.util.Scanner;

Option 2: Access Specifiers

- **Public** – accessible everywhere
- **Private** – within class only
- **Protected** – within package and subclass
- **Default** – within package

Alternative Option: Collections

- Java **Collections Framework** is a set of classes like ArrayList, HashSet, etc.

ArrayList<String> list = new ArrayList<>();

list.add("Hello");

Wrapper Classes – convert primitives to objects (Integer, Double, etc.)

int a = 5;

Integer obj = Integer.valueOf(a);

✅ Q.3 (A) One Sentence Answers

Set 1:

1. An **exception** is a problem that arises during execution.
2. A **thread** is a lightweight subprocess.
3. **Character Stream** handles character-based I/O (Reader/Writer classes).

Set 2:

1. **Byte Stream** handles byte-based I/O (InputStream/OutputStream).
2. start(), run(), sleep(), join()
3. Examples: NullPointerException, ArithmeticException

✅ Q.3 (B) Short Answers

Thread Synchronization – prevents thread interference using synchronized keyword.

File Class – used to represent file and directory pathnames.

```
File f = new File("test.txt");
```

Daemon vs Non-Daemon

- **Daemon**: runs in background (setDaemon(true)).
- **Non-Daemon**: foreground thread (main thread).

✅ Q.3 (C) Long Answers

FileWriter Example:

```
FileWriter fw = new FileWriter("file.txt");
fw.write("Hello Java");
fw.close();
```

InputStream Example:

```
FileInputStream fis = new FileInputStream("file.txt");
int i = fis.read();
System.out.print((char)i);
fis.close();
```

FileReader Example:

```
FileReader fr = new FileReader("file.txt");
int i;
while((i=fr.read())!=-1)
  System.out.print((char)i);
fr.close();
```

OutputStream Example:

```
FileOutputStream fos = new FileOutputStream("output.txt");

String str = "Data";

fos.write(str.getBytes());

fos.close();
```

✅ Q.4 (A) One Sentence Answers

Set 1:

1. Oracle
2. Application class
3. RIA = *Rich Internet Application*

Set 2:

1. init(), start(), stop()
2. JavaFX is a platform for creating GUIs
3. UI = *User Interface*

✅ Q.4 (B) Short Answers

Color Class – used to set color: Color.RED, Color.rgb(255,0,0)

Font Class – used to style text:

```
Font font = Font.font("Verdana", FontWeight.BOLD, 20);
```

JavaFX Shapes – like Rectangle, Circle, Line, etc.

Effects – visual effects like DropShadow, Glow, Reflection.

✅ Q.4 (C) Long Answers

JavaFX Architecture:

- Uses **Scene Graph**.
- Consists of: Stage, Scene, Nodes, Media, Controls.

Features of JavaFX:

- Rich GUI
- Hardware acceleration
- CSS & FXML support
- Multi-platform

JavaFX Layouts: HBox, VBox, BorderPane, GridPane

Application Structure:

```
public class MyApp extends Application {
  public void start(Stage stage) {
    stage.setTitle("Hello");
    stage.show();
  }
  public static void main(String[] args) {
    launch(args);
  }
}
```

✅ Q.5 (A) One Sentence Answers

Set 1:

1. javafx.scene.control
2. launch()
3. setScene()

Set 2:

1. javafx.scene.layout
2. ActionEvent
3. Label() and Label(String text)

✅ Q.5 (B) Short Answers

Types of Events – ActionEvent, KeyEvent, MouseEvent, etc.

Slider – used to select value by sliding a knob.

TextField – for single-line input.
TextArea – for multi-line input.

✅ Q.5 (C) Long Answers

ListView vs ComboBox:

Feature	**ListView**	**ComboBox**
Multi-select	Yes	No
Drop-down	No	Yes

JavaFX Video Code:

```
Media media = new Media(new File("video.mp4").toURI().toString());

MediaPlayer player = new MediaPlayer(media);

MediaView view = new MediaView(player);
```

Audio Code:

```
Media media = new Media(new File("audio.mp3").toURI().toString());

MediaPlayer player = new MediaPlayer(media);

player.play();
```

Checkbox vs RadioButton:

Feature	**Checkbox**	**RadioButton**
Multi-select	Yes	No (in group)
Use case	Multiple choices	Single selection

Let me know if you'd like a printable version, mock test, or MCQ practice!