CHAPTER- 2

A Trip to Objectville(EXERCISE)

1. BE the Compiler

(A)

```
class StreamingSong {
String title;
String artist;
int duration;
void play() {
System.out.println("Playing song");
}
void printDetails()
{
System.out.println("This is " + title + " by " + artist);
} }
class StreamingSongTestDrive {
public static void main(String[] args) {
StreamingSong song = new StreamingSong();
song.artist = "The Beatles";
song.title = "Come Together";
song.play();
song.printDetails();
} }
```

(B)

```
class Episode {
int seriesNumber;
int episodeNumber;
void play(){
System.out.println(“EPISODE is been played”);
}
void skipIntro() {
System.out.println("Skipping intro...");
}
void skipToNext() {
System.out.println("Loading next episode...");
}
}
class EpisodeTestDrive {
public static void main(String[] args) {
Episode episode = new Episode();
episode.seriesNumber = 4;
episode.play();
episode.skipIntro();
} }
```

2. CODE MAGNET

Java program is all scrambled up on the fridge. Can you reconstruct the code snippets to make a working Java program that produces the output listed below? Some of the curly braces fell on the floor and they were too small to pick up, so feel free to add as many of those as you need.

```
class DrumKit {
boolean topHat = true;
boolean snare = true;
void playTopHat () {
System.out.println("ding ding da-ding");
}
void playSnare() {
System.out.println("bang bang ba-bang");
}

class DrumKitTestDrive {
 public static void main(String [] args) {
 DrumKit d = new DrumKit();
 d.playSnare();
 d.playTopHat();
 d.snare = false;
   if (d.snare == true) {
      d.playSnare();
    }
 }
}
```

3. Pool Puzzle

```
public class EchoTestDrive {
public static void main(String[] args) {
Echo e1 = new Echo();
Echo e2 = new Echo();
int x = 0;
while (x < 4) {
e1.hello();
e1.count = e1.count + 1;
if (x == 3) {
e2.count = e2.count + 1;
}
if (x > 0)
{
e2.count = e2.count + e1.count;
}
 x = x + 1;
```

```
}
System.out.println(e2.count);
}
}
class Echo {
int count = 0;
void hello() {
System.out.println("helloooo... ");
}
}
```

4. Who Am I?

1. I am compiled from a .java file - **class**
2. My instance variable values can be different from my buddy's values. - **class**
3. I behave like a template. - **method**
4. I like to do stuff. - **object**
5. I can have many methods. - **class**
6. I represent "state." – **instance variable**
7. I have behaviors. - **object**
8. I am located in objects. – **instance variable**
9. I live on the heap. - **object**
10. I am used to create object instances. - **class**
11. My state can change. – **instance variable**
12. I declare methods. - **class**
13. I can change at runtime. – **instance variable**