

Q3. [Written Question] Objects and Classes [8 points]

In the following set of declarations, the second and third statements produce a compile-time error, and the remaining statements do not produce any errors. You may assume that there are no errors whatsoever in the definitions of **G** and **H**.

1	H h1 = new H (); //no error
2	H h2 = new G (); //produces a compile-time error
3	G g1 = new G (); //produces a compile-time error
4	G g2 = new H (); //no error

- Deduce the relationship between **G** and **H**.
- State their possible types (i.e. a concrete class, abstract class or interface).
- Explain your answers.

Q3 continues here

Q4. [Written] Exceptions [8 points]

```

1  public class TestExceptions {
2
3      public static void main(String[] args) {
4          int input;
5          try {
6              f(input);
7          } catch (IllegalArgumentException ex) {
8              System.out.print(____);
9          }
10     }
11
12     static void f(int x) throws IllegalArgumentException {
13
14         if (x > 10) throw new IllegalArgumentException();
15         System.out.print(____);
16         try {
17             g(x);
18         } catch (IllegalArgumentException ex) {
19             System.out.print(____);
20         }
21     }
22
23     static void g(int x) throws IllegalArgumentException {
24         System.out.print(____);
25         if (x < 0) throw new IllegalArgumentException();
26         System.out.print(____);
27         h(x);
28     }
29
30     static void h(int x) {
31         System.out.print(____);
32     }
33 }

```

Consider the code above. Two scenarios for Line 4 are given below. For each scenario, the output in the console is given below.

	Scenario 1	Scenario 2
Line 4	int input = -1;	int input = 11;
Output in console	CDE	G

Deduce what should be in the blanks above and explain your answer. Not all blanks will be filled. No credit will be given if your answers is not accompanied by an explanation.

Q4 continues here

Q5. [Written] Java Nested Classes [8 points]

```
1 public class Mark {
2
3     int z;
4
5     Mark(){
6         z = 1;
7     }
8
9     void m(){
10        System.out.println(_____);
11    }
12
13    class Foo {
14        int y;
15        static int u;
16
17        Foo(){
18            y = 2;
19            u = 2;
20        }
21
22        void f(){ System.out.println(y); }
23    }
24
25    static class Bar{
26
27        int x;
28        static int w;
29
30        Bar(){
31            x = 1;
32            w = 1;
33        }
34
35        void b(){ f(); }
36    }
37 }
```

- a. The compiler reports two errors at Line 15 and Line 35. Explain why these lines cause errors.
- b. What code should be in the blanks at Line 10 to print out
 - i. the value of **w**, first declared at Line 28?
 - ii. the value of **x**, first declared at Line 27?
 - iii. the value of **y**, first declared at Line 14 ?

Q5 continues here

Q6. [Written] Android Question [12 points]

A partially-completed RecyclerView adapter class **FlightAdapter.java** is shown below. This is meant to display flight data in ViewHolders within a RecyclerView widget. Each ViewHolder will show the Flight Number followed by the Destination. Relevant **import** statements are not shown, you may assume that they are present. The xml layout file (**flight_card.xml**) that is referenced by **FlightAdapter.java** is also given below.

FlightAdapter.java:

```
14 public class FlightAdapter extends RecyclerView.Adapter<FlightAdapter.FlightData> {
15
16     Context context;
17     LayoutInflater mlayoutInflater;
18     ArrayList<HashMap<String,String>> data;
19     final static String FLIGHT_NUMBER_KEY = "Flight";
20     final static String DESTINATION_KEY = "Destination";
21
22     FlightAdapter(Context context, ArrayList<HashMap<String, String>> data){
23         this.context = context;
24         mlayoutInflater = LayoutInflater.from(context);
25         this.data = data;
26     }
27
28     @Override public FlightData onCreateViewHolder(ViewGroup viewGroup, int i) {
29         View view = mlayoutInflater.inflate(R.layout.flight_card, viewGroup, false);
30         return new FlightData(view);
31     }
32
33     @Override public void onBindViewHolder(FlightData flightData, int i) {
34     }
35
36     @Override public int getItemCount() { return 0; }
37
38     static class FlightData extends RecyclerView.ViewHolder{
39         TextView textViewFlightNumber;
40         TextView textViewDestination;
41
42         FlightData(View view){
43             super(view);
44             textViewDestination = view.findViewById(R.id.textViewDestination);
45             textViewFlightNumber = view.findViewById(R.id.textViewFlightNumber);
46         }
47
48         void bindData(HashMap<String, String> hashMap){
49             textViewDestination.setText( hashMap.get(DESTINATION_KEY));
50             textViewFlightNumber.setText( hashMap.get(FLIGHT_NUMBER_KEY));
51         }
52     }
53 }
```

flight_card.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="wrap_content" android:layout_height="wrap_content">
4     <TextView
5         android:id="@+id/textViewFlightNumber"
6         android:text="TR123"
7         android:layout_width="wrap_content"
8         android:layout_height="wrap_content"/>
9     <TextView
10        android:id="@+id/textViewDestination"
11        android:text="Palembang"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"/>
14 </LinearLayout>
```

- a. The data to be displayed in the RecyclerView widget is passed to the adapter's constructor as an object. Describe the structure of this object as completely as you can. (4 points)
- b. When invoked, what does **onCreateViewHolder()** (Lines 28 – 31) do? Explain as fully as you can. (4 points)
- c. Complete the code for **onBindViewHolder()** (2 points)
- d. The current implementation of **getItemCount()** at Line 36 results in no data being displayed in the RecyclerView widget. Explain why and write code to rectify it. (2 points)

Q6 continues here: