1.1

Original Code:

```
class Car {
String brand;
}
// Can the main method be written like this?
public static void main(String[] args) {
Car myCar = new Car();
}
```

Optimize code:

```
class Car {
String brand;
}
public class CarExample {
public static void main(String[] args) {
Car myCar = new Car();
myCar.brand = "Honda";
System.out.println("Car brand: " + myCar.brand);
}
}
```

• What is wrong (if anything) with the given code? Explain your answer.

// Main method is floating outside any class now(Syntact Error).

• Where should the main method be located in a Java program?

// Must be inside a class and public static void main(String[] args)

• Write an example of two interacting classes in Java and explain their roles.

// AExample.java

```
class AExample {
	String brand;

	void aisa() {
		System.out.println(brand + " is a car");
	}
}
// BExample.java
public class BExample {

	public static void main(String[] args) {
		AExample myCar = new AExample();

		myCar.brand = "Honda";

		myCar.aisa();
	}
```

}
1.2
1. Review the explanations and examples above on Type Casting in Java.
Yes.

2. Explain widening (implicit casting) and narrowing (explicit casting) in your logbook.
// Widening is converting from a smaller data type to a larger data type automatically.
// Narrowing is converting from a larger data type to a smaller data type manually.

3. Summarize the key takeaways in your own words.
// Widening:
Moves from lower precision to higher precision & Happens automatically.
// Narrowing:
Moves from longer precision to lower precision & Happens manually.

4. Provide at least one example (different from the given examples) to demonstrate both implicit and explicit casting.

```
// Widening:
byte smallNumber = 67;
int mediumNumber = smallNumber;
double largeNumber = mediumNumber;
System.out.println(smallNumber);
System.out.println(mediumNumber);
System.out.println(largeNumber);
// Narrowing:
double decimalNumber = 677.666;
int wholeNumber = (int) decimalNumber;
byte tinyNumber = (byte) wholeNumber;

System.out.println(decimalNumber);
System.out.println(wholeNumber);
System.out.println(tinyNumber);
```

Source \ Destination	byte	short	int	long	float	double	char	boolean
byte	yes	yes	yes	yes	yes	yes	no	no
short	no	yes	yes	yes	yes	yes	no	no
int	no	no	yes	yes	yes	yes	no	no
long	no	no	no	yes	yes	yes	no	no
float	no	no	no	no	yes	yes	no	no
double	no	no	no	no	no	yes	no	no
char	no	no	yes	yes	yes	yes	yes	no

boolean	no	no	no	no	no	no	no	yes

1.3
Modified Code:

```
class Elephant {
	String name; // Instance variable to store elephant's name
}
public class ElephantTest {
	public static void main(String[] args) {
		Elephant a;				// Declare reference variable a
		a = new Elephant();			// Create new Elephant object , 'a' points to Object1 in memory
		a.name = "Elephant1";			// Set name of first elephant , Object1.name = "Elephant1"

		Elephant b;				// Declare reference variable b
		b = new Elephant();			// Create NEW Elephant object (different from a), , 'b' points to Object1 in memory
		b.name = "Elephant2";			// Set name of second elephant, Object1.name = "Elephant2"

		System.out.println("a is " + a.name + ", b is " + b.name);
	}
}
```

Output:

```
a is Elephant1, b is Elephant2
```

1.4
write in your logbook, an example of method.

```
public double calculateArea(double length, double width) {
	double area = length * width;    // Calculate area
	return area;
}
```

Give one example each for built-in-methods and user-defined methods

```
// Built-in-methods
String text = "Hello World";

int len = text.length();
System.out.println("Length: " + len);
```

```
// user-defined methods
public boolean isEven(int number) {
        if (number % 2 == 0) {
                return true;
        } else {
                return false;
        }
}

public static void main(String[] args) {
        MyClass obj = new MyClass();
        boolean result = obj.isEven(10);
        System.out.println("Is 10 even? " + result);
}
```

1.4.1

Method Type	Parameters (Yes/No)	Return Type (Yes/No)	Example

No Parameters, No Return Type	No	NO	public void a() {
			System.out.println("Hello!");
			}

Has Parameters, No Return Type	Yes	No	public void displayA(int a) {
			System.out.println("A: " + a);
			}

No Parameters, Has Return Type	NO	Yes	import java.util.Random;
			
			public int getRandom() {
			return (int)(Math.random() *
			100);
			}

Has Parameters, Has Return Type	Yes	Yes	public int add(int a, int b) {
			return a + b;
			}

1.4.2:

Original Code:

```
class Example {
    static void _____() { // Fill in the method name
        System.out.println("This is a method belonging to the class.");
}
    void _____() { // Fill in the method name
```

```
        System.out.println("This is a method belonging to an instance.");
}
}
public class Main {
    public static void main(String[] args) {
    _______; // Call the static method
    Example obj = new Example();
    _______; // Call the non-static method
    }
}
```

Complete Code:

```
class Example {
      static void displayClassMessage() {

            System.out.println("This is a method belonging to the class.");
      }

      void displayInstanceMessage() {
            System.out.println("This is a method belonging to an instance.");
      }
}

public class StaticMethodExample {
      public static void main(String[] args) {
            Example.displayClassMessage();

            Example obj = new Example();

            obj.displayInstanceMessage();

      }
}
```

Output:

```
This is a method belonging to the class.
This is a method belonging to an instance.
```

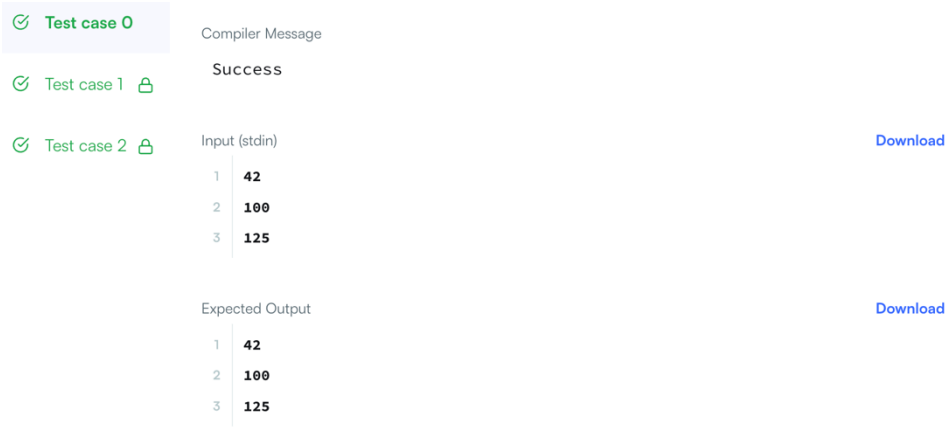
2. HackerRank Programming Exercise

2.1

Code:

```java
import java.util.*;
public class Solution {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int a = scan.nextInt();
        // Complete this line
        int b = scan.nextInt();
        // Complete this line
        int c = scan.nextInt();

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        // Complete this line
        // Complete this line
    }
}
```

Output:



2.2

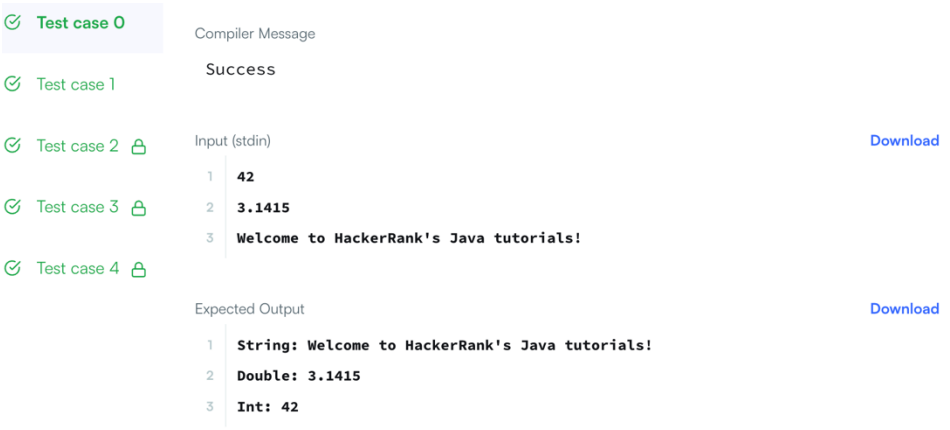
Code:

```java
import java.util.Scanner;
public class Solution {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int i = scan.nextInt();
        // Write your code here.
        double d = scan.nextDouble();
        scan.nextLine();                // consume leftover newline
        String s = scan.nextLine();

        System.out.println("String: " + s);
```

```
            System.out.println("Double: " + d);
            System.out.println("Int: " + i);
        }
}
```

```
1   import java.util.Scanner;
2
3   public class Solution {
4
5       public static void main(String[] args) {
6           Scanner scan = new Scanner(System.in);
7           int i = scan.nextInt();
8
9           // Write your code here.
10          double d = scan.nextDouble();
11          scan.nextLine();          // consume leftover newline
12          String s = scan.nextLine();
13
14          System.out.println("String: " + s);
15          System.out.println("Double: " + d);
16          System.out.println("Int: " + i);
17      }
18  }
19
```

Output:



2.3

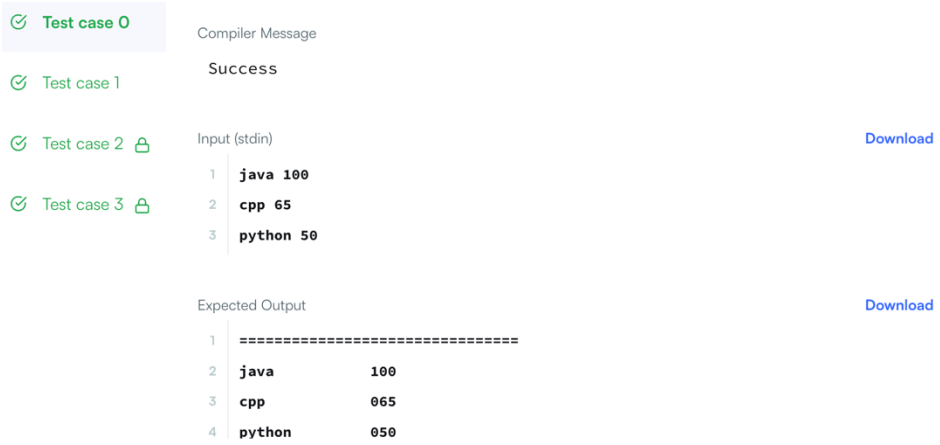
Code:

```
import java.util.Scanner;
public class Solution {
    public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            System.out.println("================================");
            for(int i=0;i<3;i++){
                String s1=sc.next();
```

```
                int x=sc.nextInt();
                //Complete this line
                System.out.printf("%-15s%03d%n",s1,x);

            }
            System.out.println("================================");
    }
}
```

```
import java.util.Scanner;

public class Solution {

    public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            System.out.println("================================");
            for(int i=0;i<3;i++){
                String s1=sc.next();
                int x=sc.nextInt();
                //Complete this line
                System.out.printf("%-15s%03d%n",s1,x);

            }
            System.out.println("================================");

    }
}
```

Output:



2.4

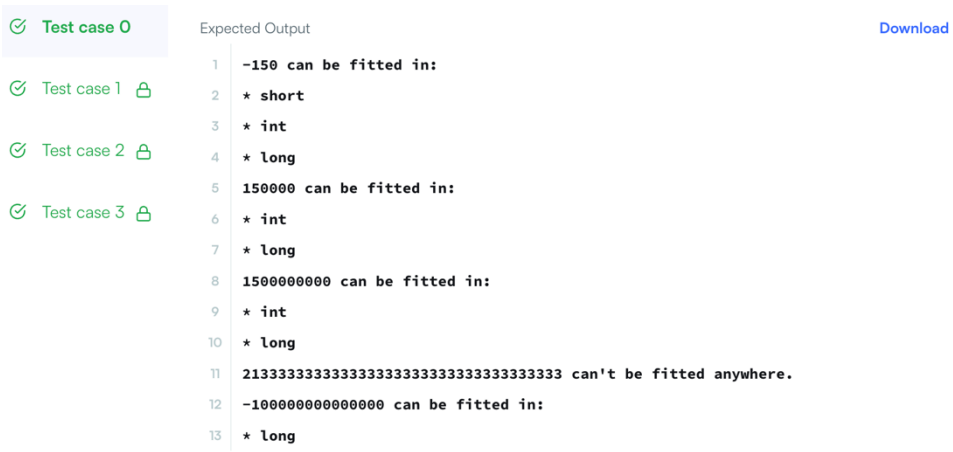
Code:

```
import java.util.*;
import java.io.*;
class Solution{
    public static void main(String []argh)
    {
```

```
        Scanner sc = new Scanner(System.in);
        int t=sc.nextInt();
        for(int i=0;i<t;i++)
        {
            try
            {
                long x=sc.nextLong();
                System.out.println(x+" can be fitted in:");
                if(x>=-128 && x<=127)System.out.println("* byte");
                //Complete the code
                if (x >= Short.MIN_VALUE && x <= Short.MAX_VALUE)
                    System.out.println("* short");
                if (x >= Integer.MIN_VALUE && x <= Integer.MAX_VALUE)
                    System.out.println("* int");
                if (x >= Long.MIN_VALUE && x <= Long.MAX_VALUE)
                    System.out.println("* long");
            }
            catch(Exception e)
            {
                System.out.println(sc.next()+" can't be fitted anywhere.");
            }
        }
    }
}
```

```java
class Solution{
    public static void main(String []argh)
    {
        Scanner sc = new Scanner(System.in);
        int t=sc.nextInt();

        for(int i=0;i<t;i++)
        {
            try
            {
                long x=sc.nextLong();
                System.out.println(x+" can be fitted in:");
                if(x>=-128 && x<=127)System.out.println("* byte");
                //Complete the code
                if (x >= Short.MIN_VALUE && x <= Short.MAX_VALUE)
                    System.out.println("* short");

                if (x >= Integer.MIN_VALUE && x <= Integer.MAX_VALUE)
                    System.out.println("* int");

                if (x >= Long.MIN_VALUE && x <= Long.MAX_VALUE)
                    System.out.println("* long");
            }
            catch(Exception e)
            {
                System.out.println(sc.next()+" can't be fitted anywhere.");
            }

        }
    }
}
```

Output:



2.5

Code:

```java
import java.util.*;
import java.text.*;
public class Solution {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double payment = scanner.nextDouble();
```

```
        scanner.close();
        NumberFormat usFormat =
NumberFormat.getCurrencyInstance(Locale.US);
        NumberFormat indiaFormat = NumberFormat.getCurrencyInstance(new
Locale("en", "IN"));
        NumberFormat chinaFormat =
NumberFormat.getCurrencyInstance(Locale.CHINA);
        NumberFormat franceFormat =
NumberFormat.getCurrencyInstance(Locale.FRANCE);
        String us = usFormat.format(payment);
        String india = indiaFormat.format(payment);
        String china = chinaFormat.format(payment);
        String france = franceFormat.format(payment);
        System.out.println("US: " + us);
        System.out.println("India: " + india);
        System.out.println("China: " + china);
        System.out.println("France: " + france);
    }
}
```

```java
import java.util.*;
import java.text.*;

public class Solution {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double payment = scanner.nextDouble();
        scanner.close();

        NumberFormat usFormat = NumberFormat.getCurrencyInstance(Locale.US);
        NumberFormat indiaFormat = NumberFormat.getCurrencyInstance(new Locale("en", "IN"));
        NumberFormat chinaFormat = NumberFormat.getCurrencyInstance(Locale.CHINA);
        NumberFormat franceFormat = NumberFormat.getCurrencyInstance(Locale.FRANCE);

        String us = usFormat.format(payment);
        String india = indiaFormat.format(payment);
        String china = chinaFormat.format(payment);
        String france = franceFormat.format(payment);

        System.out.println("US: " + us);
        System.out.println("India: " + india);
        System.out.println("China: " + china);
        System.out.println("France: " + france);
    }
}
```

Output:

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5
- Test case 6

Compiler Message

Success

Input (stdin) Download

```
12324.134
```

Expected Output Download

```
US: $12,324.13
India: Rs.12,324.13
China: ￥12,324.13
France: 12 324,13 €
```