Review Questions

Typecasting

1- Can, or not?

```
jshell> int i;
i ==> 0
jshell> double d;
d ==> 0.0
jshell> i = d;
|  Error:
|  incompatible types: possible lossy conversion from double to int
|      i = d;
|          ^
jshell> d = i;
d ==> 0.0
jshell> i = (int) d;
i ==> 0
jshell> d = (double) i;
d ==> 0.0
```

2- Can, or not?

```
jshell> int i;
i ==> 0
jshell> boolean b;
b ==> false
jshell> i = b;
|  Error:
|  incompatible types: boolean cannot be converted to int
|      i = b;
|          ^
jshell> b = i;
|  Error:
|  incompatible types: int cannot be converted to boolean
|      b = i;
|          ^
jshell> i = (int) b;
|  Error:
|  incompatible types: boolean cannot be converted to int
|      i = (int) b;
|                ^
jshell> b = (boolean) i;
```

```
|  Error:
|  incompatible types: int cannot be converted to boolean
|      b = (boolean) i;
|                   ^
```

3- Can, or not?

```
jshell> class A {
   ...> }
|  created class A
jshell> class B extends A {
   ...> }
|  created class B
jshell> A a = new B();
a ==> B@5702b3b1
jshell> B b = new A();
|  Error:
|  incompatible types: A cannot be converted to B
|      B b = new A();
|            ^-----^
```

4- Can, or not?

```
jshell> class A {
   ...> }
|  created class A
jshell> class B extends A {
   ...> }
|  created class B
jshell> A a = new A();
a ==> A@25bbe1b6
jshell> B b = new B();
b ==> B@69ea3742
jshell> b = (B)a;
|  java.lang.ClassCastException thrown: REPL.$JShell$11$A cannot be cast to REPL.$JShell$1
|        at (#5:1)
jshell> a = (A)b;
a ==> B@69ea3742
```

5- Can, or not?

```
jshell> interface I {
   ...> }
|  created interface I
jshell> class A implements I {
   ...> }
|  created class A
jshell> I i1 = new I();
|  Error:
|  I is abstract; cannot be instantiated
|      I i1 = new I();
```

```
|             ^-----^
jshell> I i2 = new A();
i2 ==> A@5702b3b1
jshell> A a1 = i2;
|  Error:
|  incompatible types: I cannot be converted to A
|      A a1 = i2;
|             ^^
jshell> A a2 = (A)i2;
a2 ==> A@5702b3b1
```

6- Can, or not?

```
jshell> interface I {
   ...> }
|  created interface I
jshell> interface J extends I {
   ...> }
|  created interface J
jshell> class A implements J {
   ...> }
|  created class A
jshell> A a = new A();
a ==> A@4b952a2d
jshell> I i = a;
i ==> A@4b952a2d
jshell> J j = a;
j ==> A@4b952a2d
jshell> i = j;
i ==> A@4b952a2d
jshell> j = i;
|  Error:
|  incompatible types: I cannot be converted to J
|      j = i;
|          ^
jshell> j = (J)i;
j ==> A@4b952a2d
jshell> a = i;
|  Error:
|  incompatible types: I cannot be converted to A
|      a = i;
|          ^
jshell> a = j;
|  Error:
|  incompatible types: J cannot be converted to A
|      a = j;
|          ^
jshell> a = (A)i;
a ==> A@4b952a2d
jshell> a = (A)j;
a ==> A@4b952a2d
```

7- Can, or not?

```
jshell> interface I {
   ...> }
|  created interface I
jshell> interface J {
   ...> }
|  created interface J
jshell> class A implements I, J {
   ...> }
|  created class A
jshell> A a = new A();
a ==> A@4b952a2d
jshell> I i = a;
i ==> A@4b952a2d
jshell> J j = a;
j ==> A@4b952a2d
jshell> i = j;
|  Error:
|  incompatible types: J cannot be converted to I
|      i = j;
|          ^
jshell> j = i;
|  Error:
|  incompatible types: I cannot be converted to J
|      j = i;
|          ^
jshell> j = (J)i;
j ==> A@4b952a2d
jshell> I = (I)j;
|  Error:
|  cannot find symbol
|    symbol:   variable I
|      I = (I)j;
|      ^
jshell> a = i;
|  Error:
|  incompatible types: I cannot be converted to A
|      a = i;
|          ^
jshell> a = j;
|  Error:
|  incompatible types: J cannot be converted to A
|      a = j;
|          ^
jshell> a = (A)i;
a ==> A@4b952a2d
jshell> a = (A)j;
a ==> A@4b952a2d
```

8- Can, or not?

```
jshell> class A {
   ...> }
|  created class A
```

```
jshell> class B extends A {
   ...> }
|  created class B
jshell> class C extends A {
   ...> }
|  created class C
jshell> B b = new B();
b ==> B@4b952a2d
jshell> A a = b;
a ==> B@4b952a2d
jshell> C c = b;
|  Error:
|  incompatible types: B cannot be converted to C
|      C c = b;
|            ^
jshell> A a = (A)b;
a ==> B@4b952a2d
jshell> C c = (C)b;
|  Error:
|  incompatible types: B cannot be converted to C
|      C c = (C)b;
|              ^
```

Inheritance/Method Overriding

9- Can, or not? If can, print what?

```
jshell> class A {
   ...>   void f() { System.out.println("A f"); }
   ...> }
|  created class A
jshell> class B extends A {
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@5702b3b1
jshell> b.f();
A f
jshell> A a = b;
a ==> B@5702b3b1
jshell> a.f();
A f
```

10- Can, or not? If can, print what?

```
jshell> class A {
   ...>   void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
```

```
jshell> class B extends A {
   ...>   void f() {
   ...> System.out.println("B f");
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@25bbe1b6
jshell> b.f();
B f
jshell> A a = b;
a ==> B@25bbe1b6
jshell> a.f();
B f
jshell> a = new A();
a ==> A@73846619
jshell> a.f();
A f
```

11- Can, or not? If can, print what?

```
jshell> class A {
   ...>   void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
jshell> class B extends A {
   ...>   void f() {
   ...> super.f();
   ...> System.out.println("B f");
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@5702b3b1
jshell> b.f();
A f
B f
jshell> A a = b;
a ==> B@5702b3b1
jshell> a.f();
A f
B f
```

12- Can, or not? If can, print what?

```
jshell> class A {
   ...>   void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
```

```
jshell> class B extends A {
   ...>   void f() {
   ...> this.f();
   ...> System.out.println("B f");
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@5702b3b1
jshell> b.f();
|  java.lang.StackOverflowError thrown:
|        at B.f (#2:3)
jshell> A a = b;
a ==> B@5702b3b1
jshell> a.f();
|  java.lang.StackOverflowError thrown:
|        at B.f (#2:3)
```

13- Can, or not? If can, print what?

```
jshell> class A {
   ...>   void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
jshell> class B extends A {
   ...>   int f() {
   ...> System.out.println("B f");
   ...> return 0;
   ...>   }
   ...> }
|  Error:
|  f() in B cannot override f() in A
|    return type int is not compatible with void
|        int f() {
|        ^--------...
jshell> B b = new B();
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|      ^
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|                ^
jshell> b.f();
|  Error:
|  cannot find symbol
|    symbol:   variable b
|      b.f();
|      ^
```

```
jshell> A a = b;
|  Error:
|  cannot find symbol
|    symbol:   variable b
|      A a = b;
|            ^
jshell> a.f();
|  Error:
|  cannot find symbol
|    symbol:   variable a
|      a.f();
|      ^
```

14- Can, or not? If can, print what?

```
jshell> class A {
   ...>   void f() {
   ...>     System.out.println("A f");
   ...>   }
   ...> }
|  created class A

jshell> class B extends A {
   ...>   int f(int x) {
   ...>     System.out.println("B f");
   ...>     return x;
   ...>   }
   ...> }
|  created class B

jshell> B b = new B();
b ==> B@4bec1f0c

jshell> b.f();
A f

jshell> b.f(0);
B f
$5 ==> 0

jshell> A a = b;
a ==> B@4bec1f0c

jshell> a.f();
A f

jshell> a.f(0);
|  Error:
|  method f in class A cannot be applied to given types;
|    required: no arguments
|    found: int
|    reason: actual and formal argument lists differ in length
|      a.f(0);
|      ^-^
```

15- Can, or not? If can, what will be printed?

```
jshell> class A {
   ...>   public void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public void f() {
   ...> System.out.println("B f");
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@4bec1f0c
jshell> A a = b;
a ==> B@4bec1f0c
jshell> a.f();
B f
jshell> b.f();
B f
```

16- Can, or not? If can, what will be printed?

```
jshell> class A {
   ...>   private void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public void f() {
   ...> System.out.println("B f");
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@29ca901e
jshell> A a = b;
a ==> B@29ca901e
jshell> a.f();
|  Error:
|  f() has private access in A
|      a.f();
|    ^-^
jshell> b.f();
B f
```

17- *Deleted question - duplicate of Q16*

18- Can, or not? If can, what will be printed?

```
jshell> class A {
   ...>   static void f() {
   ...> System.out.println("A f");
   ...>   }
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public void f() {
   ...> System.out.println("B f");
   ...>   }
   ...> }
|  Error:
|  f() in B cannot override f() in A
|    overridden method is static
|        public void f() {
|        ^----------------...
jshell> B b = new B();
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|   ^
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|            ^
jshell> A a = b;
|  Error:
|  cannot find symbol
|    symbol:   variable b
|      A a = b;
|         ^
jshell> a.f();
|  Error:
|  cannot find symbol
|    symbol:   variable a
|      a.f();
|   ^
jshell> b.f();
|  Error:
|  cannot find symbol
|    symbol:   variable b
|      b.f();
|   ^
```

19- Can, or not? If can, what will be printed?

```
jshell> class A {
   ...>   static void f() {
   ...> System.out.println("A f");
```

```
...>   }
...> }
|  created class A
jshell> class B extends A {
...>   static void f() {
...> System.out.println("B f");
...>   }
...> }
|  created class B
jshell> B b = new B();
b ==> B@4bec1f0c
jshell> A a = b;
a ==> B@4bec1f0c
jshell> A.f();
A f
jshell> B.f();
B f
jshell> a.f();
A f
jshell> b.f();
B f
```

20- Will the following code compile? Why?

```
jshell> class A {
   ...>   public void f(int x) {}
   ...>   public void f(boolean y) {}
   ...> }
|  created class A
```

21- Will the following code compile? Why?

```
jshell> class A {
   ...>   public void f(int x) {}
   ...>   public void f(int y) {}
   ...> }
|  Error:
|  method f(int) is already defined in class A
|        public void f(int y) {}
|     ^----------------------^
```

22- Will the following code compile? Why?

```
jshell> class A {
   ...>   private void f(int x) {}
   ...>   public void f(int y) {}
   ...> }
|  Error:
|  method f(int) is already defined in class A
|        public void f(int y) {}
|     ^----------------------^
```

23- Will the following code compile? Why?

```
jshell> class A {
   ...>   public int f(int x) {
   ...>   return x;
   ...>   }
   ...>   public void f(int y) {}
   ...> }
|  Error:
|  method f(int) is already defined in class A
|        public void f(int y) {}
|     ^--------------------^
```

24- Will the following code compile? Why?

```
jshell> class A {
   ...>   public void f(int x, String s) {}
   ...>   public void f(String s, int y) {}
   ...> }
|  created class A
```

25- Will the following code compile? Why?

```
jshell> class A {
   ...>   public void f(int x) {}
   ...>   public void f(int y) throws IOException {}
   ...> }
|  Error:
|  method f(int) is already defined in class A
|        public void f(int y) throws IOException {}
|     ^---------------------------------------^
```

26- Will the following code compile? If so, what will be printed?

```
jshell> class A {
   ...>   private int x = 0;
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public void f() {
   ...> System.out.println(x);
   ...>   }
   ...> }
|  Error:
|  x has private access in A
|          System.out.println(x);
|                             ^
jshell> B b = new B();
|  Error:
|  cannot find symbol
|    symbol:   class B
```

```
|      B b = new B();
|    ^
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|            ^
jshell> b.f();
|  Error:
|  cannot find symbol
|    symbol:   variable b
|      b.f();
|    ^
```

27- Will the following code compile? If so, what will be printed?

```
jshell> class A {
   ...>   private int x = 0;
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public void f() {
   ...> System.out.println(super.x);
   ...>   }
   ...> }
|  Error:
|  x has private access in A
|          System.out.println(super.x);
|                      ^-----^
jshell> B b = new B();
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|    ^
|  Error:
|  cannot find symbol
|    symbol:   class B
|      B b = new B();
|            ^
jshell> b.f();
|  Error:
|  cannot find symbol
|    symbol:   variable b
|      b.f();
|    ^
```

28- Will the following code compile? If so, what will be printed?

```
jshell> class A {
   ...>   protected int x = 0;
   ...> }
|  created class A
```

```
jshell> class B extends A {
   ...>   public void f() {
   ...> System.out.println(x);
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@4bec1f0c
jshell> b.f();
0
```

29- Will the following code compile? If so, what will be printed?

```
jshell> class A {
   ...>   protected int x = 0;
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public int x = 1;
   ...>   public void f() {
   ...> System.out.println(x);
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@4bec1f0c
jshell> b.f();
1
```

30- Will the following code compile? If so, what will be printed?

```
jshell> class A {
   ...>   protected int x = 0;
   ...> }
|  created class A
jshell> class B extends A {
   ...>   public int x = 1;
   ...>   public void f() {
   ...> System.out.println(super.x);
   ...>   }
   ...> }
|  created class B
jshell> B b = new B();
b ==> B@29ca901e
jshell> b.f();
0
```

Exceptions

31- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
   ...>   static void f() throws IllegalArgumentException {
   ...> try {
   ...>   System.out.println("Before throw");
   ...>   throw new IllegalArgumentException();
   ...>   System.out.println("After throw");
   ...> } catch (IllegalArgumentException e) {
   ...>   System.out.println("Caught in f");
   ...> }
   ...>   }
   ...>   public static void main(String[] args) {
   ...> try {
   ...>   System.out.println("Before f");
   ...>   f();
   ...>   System.out.println("After f");
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught in main");
   ...> }
   ...>   }
   ...> }
|  Error:
|  unreachable statement
|            System.out.println("After throw");
|            ^-------------------------------^
```

32- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
   ...>   static void f() throws IllegalArgumentException {
   ...> try {
   ...>   throw new IllegalArgumentException();
   ...> } catch (IllegalArgumentException e) {
   ...>   System.out.println("Caught in f");
   ...> }
   ...>   }
   ...>   public static void main(String[] args) {
   ...> try {
   ...>   System.out.println("Before f");
   ...>   f();
   ...>   System.out.println("After f");
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught in main");
   ...> }
   ...>   }
   ...> }
|  created class Main
Before f
Caught in f
After f
```

33- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
   ...>   static void f() throws IllegalArgumentException {
   ...> try {
   ...>   throw new Exception();
   ...> } catch (IllegalArgumentException e) {
   ...>   System.out.println("Caught in f");
   ...> }
   ...>   }
   ...>   public static void main(String[] args) {
   ...> try {
   ...>   System.out.println("Before f");
   ...>   f();
   ...>   System.out.println("After f");
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught in main");
   ...> }
   ...>   }
   ...> }
|  Error:
|  unreported exception java.lang.Exception; must be caught or declared to be thrown
|            throw new Exception();
|            ^--------------------^
```

34- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
   ...>   static void f() throws Exception {
   ...> try {
   ...>   throw new IllegalArgumentException();
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught in f");
   ...> }
   ...>   }
   ...>   public static void main(String[] args) {
   ...> try {
   ...>   System.out.println("Before f");
   ...>   f();
   ...>   System.out.println("After f");
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught in main");
   ...> }
   ...>   }
   ...> }
|  created class Main
Before f
Caught in f
After f
```

35- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
   ...>   static void f() throws Exception {
   ...> try {
```

```
    ...>   throw new ArrayIndexOutOfBoundsException();
    ...> } catch (IllegalArgumentException e) {
    ...>   System.out.println("Caught in f");
    ...> }
    ...>   }
    ...>   public static void main(String[] args) {
    ...> try {
    ...>   System.out.println("Before f");
    ...>   f();
    ...>   System.out.println("After f");
    ...> } catch (Exception e) {
    ...>   System.out.println("Caught in main");
    ...> }
    ...>   }
    ...> }
|  created class Main
Before f
Caught in main
```

36- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
    ...>   static void f() throws Exception {
    ...> try {
    ...>   throw new ArrayIndexOutOfBoundsException();
    ...> } catch (IllegalArgumentException e) {
    ...>   System.out.println("Caught IA exception in f");
    ...> } catch (ArrayIndexOutOfBoundsException e) {
    ...>   System.out.println("Caught AIOOB exception in f");
    ...> }
    ...>   }
    ...>   public static void main(String[] args) {
    ...> try {
    ...>   System.out.println("Before f");
    ...>   f();
    ...>   System.out.println("After f");
    ...> } catch (Exception e) {
    ...>   System.out.println("Caught in main");
    ...> }
    ...>   }
    ...> }
|  created class Main
Before f
Caught AIOOB exception in f
After f
```

37- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
    ...>   static void f() throws Exception {
    ...> try {
    ...>   throw new ArrayIndexOutOfBoundsException();
    ...> } catch (Exception e) {
```

```
...>   System.out.println("Caught exception in f");
...> } catch (ArrayIndexOutOfBoundsException e) {
...>   System.out.println("Caught AIOOB exception in f");
...> }
...>   }
...>   public static void main(String[] args) {
...> try {
...>   System.out.println("Before f");
...>   f();
...>   System.out.println("After f");
...> } catch (Exception e) {
...>   System.out.println("Caught in main");
...> }
...>   }
...> }
|  Error:
|  exception java.lang.ArrayIndexOutOfBoundsException has already been caught
|          } catch (ArrayIndexOutOfBoundsException e) {
|            ^-----------------------------------------...
```

38- Will the following code compile? If so, what will be printed?

```
jshell> class Main {
   ...>   static void f() throws Exception {
   ...> try {
   ...>   throw new ArrayIndexOutOfBoundsException();
   ...> } catch (ArrayIndexOutOfBoundsException e) {
   ...>   System.out.println("Caught AIOOB exception in f");
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught exception in f");
   ...> }
   ...>   }
   ...>   public static void main(String[] args) {
   ...> try {
   ...>   System.out.println("Before f");
   ...>   f();
   ...>   System.out.println("After f");
   ...> } catch (Exception e) {
   ...>   System.out.println("Caught in main");
   ...> }
   ...>   }
   ...> }
|  created class Main
Before f
Caught AIOOB exception in f
After f
```

Auto Boxing and Unboxing

39- Will the following code compile? If so, what will be printed?

```
jshell> List<Integer> list = new ArrayList<>();
list ==> []
jshell> int one = 1;
one ==> 1
jshell> Integer two = 2;
two ==> 2
jshell> list.add(one);
$4 ==> true
jshell> list.add(two);
$5 ==> true
jshell> list.add(3);
$6 ==> true
jshell> for (Integer num : list) {
   ...>   System.out.println(num);
   ...> }
1
2
3
```

40- Will the following code compile? If so, what will be printed?

```
jshell> List<Integer> list = new ArrayList<>();
list ==> []
jshell> int one = 1;
one ==> 1
jshell> Integer two = 2;
two ==> 2
jshell> list.add(one);
$4 ==> true
jshell> list.add(two);
$5 ==> true
jshell> list.add(3);
$6 ==> true
jshell> for (int num : list) {
   ...>   System.out.println(num);
   ...> }
1
2
3
```

41- Will the following code compile? If so, what will be printed?

```
jshell> List<Integer> list = Arrays.asList(1, 2, 3);
list ==> [1, 2, 3]
jshell> for (Double num : list) {
   ...>   System.out.println(num);
   ...> }
|  Error:
|  incompatible types: java.lang.Integer cannot be converted to java.lang.Double
|      for (Double num : list) {
|                        ^--^
```

42- Will the following code compile? If so, what will be printed?

```
jshell> List<Integer> list = Arrays.asList(1, 2, 3);
list ==> [1, 2, 3]
jshell> for (double num : list) {
   ...>   System.out.println(num);
   ...> }
1.0
2.0
3.0
```

43- Will the following code compile? If so, what will be printed?

```
jshell> double d = 5;
d ==> 5.0
jshell> int i = 2.5;
|  Error:
|  incompatible types: possible lossy conversion from double to int
|      int i = 2.5;
|              ^-^
jshell> System.out.println(d);
5.0
jshell> System.out.println(i);
|  Error:
|  cannot find symbol
|    symbol:   variable i
|      System.out.println(i);
|                         ^
```

44- Will the following code compile? If so, what will be printed?

```
jshell> double d = (int) 5;
d ==> 5.0
jshell> int i = (double) 2.5;
|  Error:
|  incompatible types: possible lossy conversion from double to int
|      int i = (double) 2.5;
|              ^----------^
jshell> System.out.println(d);
5.0
jshell> System.out.println(i);
|  Error:
|  cannot find symbol
|    symbol:   variable i
|      System.out.println(i);
|                         ^
```

45- Will the following code compile? If so, what will be printed?

```
jshell> double d = (int) 5.5;
d ==> 5.0
```

```
jshell> int i = (int) 2.5;
i ==> 2
jshell> System.out.println(d);
5.0
jshell> System.out.println(i);
2
```

46- Will the following code compile? If so, what will be printed?

```
jshell> Double d = 5;
|  Error:
|  incompatible types: int cannot be converted to java.lang.Double
|      Double d = 5;
|                 ^
jshell> Integer i = 2.5;
|  Error:
|  incompatible types: double cannot be converted to java.lang.Integer
|      Integer i = 2.5;
|                  ^-^
jshell> System.out.println(d);
|  Error:
|  cannot find symbol
|    symbol:   variable d
|      System.out.println(d);
|                         ^
jshell> System.out.println(i);
|  Error:
|  cannot find symbol
|    symbol:   variable i
|      System.out.println(i);
|                         ^
```

47- Will the following code compile? If so, what will be printed?

```
jshell> Double d = (double) 5;
d ==> 5.0
jshell> Integer i = (int) 2.5;
i ==> 2
jshell> System.out.println(d);
5.0
jshell> System.out.println(i);
2
```

48- Will the following code compile? If so, what will be printed?

```
jshell> double d = (Integer) 5;
d ==> 5.0
jshell> int i = (Integer) 2;
i ==> 2
jshell> System.out.println(d);
5.0
```

```
jshell> System.out.println(i);
2
```

49- Will the following code compile? If so, what will be printed?

```
jshell> double d = (Double) 5;
|  Error:
|  incompatible types: int cannot be converted to java.lang.Double
|      double d = (Double) 5;
|                          ^
jshell> int i = (Integer) 2;
i ==> 2
jshell> System.out.println(d);
|  Error:
|  cannot find symbol
|    symbol:   variable d
|      System.out.println(d);
|                         ^
jshell> System.out.println(i);
2
```

Generics and Collections

50- Will the following code compile? If so, what will be printed?

```
jshell> List<Integer> list = new LinkedList<>();
list ==> []
jshell> list.add(5);
$2 ==> true
jshell> list.add(4);
$3 ==> true
jshell> list.add(3);
$4 ==> true
jshell> list.add(2);
$5 ==> true
jshell> list.add(1);
$6 ==> true
jshell> Iterator<Integer> it = list.iterator();
it ==> java.util.LinkedList$ListItr@20322d26
jshell> while (it.hasNext()) {
   ...>   System.out.println(it.next());
   ...> }
5
4
3
2
1
```

51- Will the following code compile? If so, what will be printed?

```
jshell> ArrayList<Integer> list = new ArrayList<>();
list ==> []
jshell> list.add(5);
$2 ==> true
jshell> list.add(4);
$3 ==> true
jshell> list.add(3);
$4 ==> true
jshell> list.add(2);
$5 ==> true
jshell> list.add(1);
$6 ==> true
jshell> Collections.sort(list);
jshell> for (int i : list) {
   ...>   System.out.println(i);
   ...> }
1
2
3
4
5
```

52- Will the following code compile? If so, what will be printed?

```
jshell> List<Integer> list = Arrays.asList(1, 2, 4, 4, 5);
list ==> [1, 2, 4, 4, 5]
jshell> Collections.sort(list, new Comparator<>() {
   ...>   @Override
   ...>   public int compare(Integer i1, Integer i2) {
   ...> return -i1.compareTo(i2);
   ...>   }
   ...> });
jshell> list.forEach(System.out::println);
5
4
4
2
1
```

53- Will the following code compile? If so, what will be printed?

```
jshell> Set<Integer> set = new HashSet<>(Arrays.asList(5, 2, 4, 1, 4, 2));
set ==> [1, 2, 4, 5]
jshell> set.forEach(System.out::println);
1
2
4
5
```

54- Will the following code compile? If so, what will be printed?

```
jshell> Map<Integer, String> map = new HashMap<>();
map ==> {}
jshell> map.put(2, "world");
$2 ==> null
jshell> map.put(2, "cs2030");
$3 ==> "world"
jshell> map.put(1, "hello");
$4 ==> null
jshell> for (Map.Entry<Integer, String> entry : map.entrySet()) {
   ...>   System.out.println(entry.getKey() + ": " + entry.getValue());
   ...> }
1: hello
2: cs2030
```

55- Will the following code compile? If so, what will be printed?

```
jshell> Map<Integer, String> map = new HashMap<>();
map ==> {}
jshell> map.put(1, "bell");
$2 ==> null
jshell> map.put(2, "curve");
$3 ==> null
jshell> map.put(9001, "god");
$4 ==> null
jshell> map.forEach((k, v) -> System.out.println(k + ": " + v));
1: bell
2: curve
9001: god
```

56- Will the following code compile? If so, what will be printed?

```
jshell> Map<Integer, String> map = new HashMap<>();
map ==> {}
jshell> map.put(2, "bell");
$2 ==> null
jshell> map.put(1, "curve");
$3 ==> null
jshell> map.put(9001, "god");
$4 ==> null
jshell> map.forEach((k, v) -> System.out.println(k + ": " + v));
1: curve
2: bell
9001: god
```

57- Will the following code compile? If so, what will be printed?

```
jshell> Map<Integer, String> map = new HashMap<>();
map ==> {}
jshell> map.put(10, "bell");
$2 ==> null
jshell> map.put(1, "curve");
```