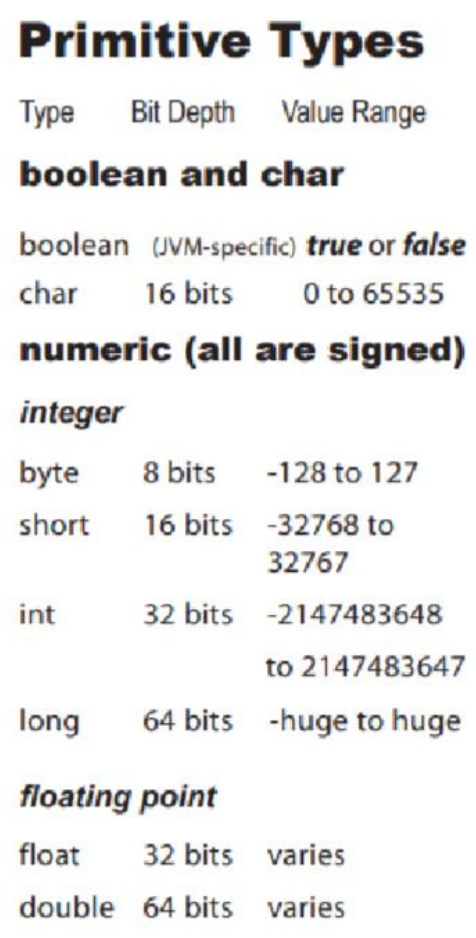
CHAPTER 3

Primitives and References(Exercise)

PRIMITIVE TYPES



Primitive Types

Type	Bit Depth	Value Range
boolean and char		
boolean	(JVM-specific)	***true*** or ***false***
char	16 bits	0 to 65535
numeric (all are signed)		
integer		
byte	8 bits	-128 to 127
short	16 bits	-32768 to 32767
int	32 bits	-2147483648 to 2147483647
long	64 bits	-huge to huge
floating point		
float	32 bits	varies
double	64 bits	varies

EXERCISE

1. BE the Compiler

(A)

```
class Books {
String title; String author;
}
class BooksTestDrive {
public static void main(String[] args) {
Books[] myBooks = new Books[3];
int x = 0;
myBooks[0] = new Books();
myBooks[1] = new Books();
myBooks[2] = new Books();
myBooks[0].title = "The Grapes of Java";
myBooks[1].title = "The Java Gatsby";
myBooks[2].title = "The Java Cookbook";
myBooks[0].author = "bob";
myBooks[1].author = "sue";
myBooks[2].author = "ian";
while (x < 3) {
System.out.print(myBooks[x].title);
System.out.print(" by ");
System.out.println(myBooks[x].author);
x = x + 1;
} } }
```

(B)

```
class Hobbits {
String name;
public static void main(String[] args) {
Hobbits[] h = new Hobbits[3];
int z = -1;
while (z < 2) {
z = z + 1;
h[z] = new Hobbits();
h[z].name = "bilbo";
if (z == 0) {
h[z].name = "frodo";
}
if (z == 1) {
h[z].name = "sam";
}
System.out.print(h[z].name + " is a ");
System.out.println("good Hobbit name");
} } }
```

2. Code Magnets

```
class TestArrays {
public static void main(String [] args) {
int y = 0;
int ref;
int [] index = new int[4];
index[0] = 1;
index[1] = 3;
index[2] = 0;
index[3] = 2;
String [] islands = new String[4];
islands[0] = "Bermuda";
islands[1] = "Fiji";
islands[2] = "Azores";
islands[3] = "Cozumel";
while (y < 4) {
ref = index[y];
System.out.print("island = ");
System.out.println(islands[ref]);
y = y + 1;
}
}
```

3.POOL PUZZLE

```
class Triangle {
   double area;
   int height;
   int length;
   public static void main(String[] args) {
      int x = 0;
      Triangle[] ta = new Triangle[4];
      while (x < 4) {
         ta[x] = new Triangle();
         ta[x].height = (x + 1) * 2;
         ta[x].length = x + 4;
         ta[x].setArea();
         System.out.print("triangle " + x + ", area");
         System.out.println(" = " + ta[x].area);
         x = x + 1;
      }
      int y = x;
      x = 27;
      Triangle t5 = ta[2];
```

```
        ta[2].area = 343;
        System.out.print("y = " + y);
        System.out.println(", t5 area = " + t5.area);
    }
    void setArea() {
        area = (height * length) / 2.0;
    }
}
```

OUTPUT
y=4, t5 area = 343.0

4. A Heap o' Trouble

Ans-
h[0]=null;
h[1]=id 1;
h[2]=null;
h[3]=id 2;
h[4]=id 0;

5. Five-Minute Mystery

Ans-
Because Kate's code was a complete mess. She was not creating the reference variable for all the contacts and so that none can be used except the last one out of ten. While Bob's method was perfect such that each and every contact can be accessed.