Chapter 2 – Objects and Primitive Data - AP

Chapter Objectives

- Define the difference between primitive data and objects.
- Declare and use variables.
- Perform mathematical computations.
- Create objects and use them.
- Explore the difference between a Java application and a Java applet.
- Create graphical programs that draw shapes.

Chapter Overview: During this chapter you should begin to feel comfortable with the declaration of variables, and discuss what happens at each point in declaring an object, creating an instance of that object, assigning vs. changing the state of the object and how that differs from what happens with primitive data. You will begin to do mathematical calculations and should come to understand both the mathematical operators and their precedence. Concept and syntax of enumerated data types is introduced which is an important AP topic.

Multiple Choice: 2.1 – 2.10

True False: 2.1 – 2.8

Short Answer: 2.1 – 2.7, 2.9 – 2.14

AP Multiple Choice: 2.1 – 2.3

Worksheets: Data Type Identifiers, Precedence, String Methods

Programming Projects: 2.1, 2.3, 2.4, 2.7, 2.8, 2.13

PRIMITIVE DATA TYPES IDENTIFIERS IN JAVA

1. State whether the following are legal identifiers in Java. If they are not legal, indicate why.

    a. number
    b. 5number
    c. thisNumber
    d. that_number
    e. big number
    f. char
    g. character

2. State whether the following are legal declarations and initializations of identifiers. If they are not legal, indicate how to correct them.

    a. **int** num1 = 8;
    b. **int** num2 = 7.6;
    c. **float** num3 = 5;
    d. **float** num4 = 5.9;
    e. **float** num5 = 4.34e5;
    f. **char** ch1 = ‘v’;
    g. **char** ch2 = 84;
    h. **double** num6 = 78456;
    i. **double** num7 = 23,567;
    j. **double** num8 = $123.68;
    k. **long** num9 = 28;

3. Use the Unicode chart on page 576 find the output of the following code:

```
char letter1 = 'J';
char letter2 = 'a';
char letter3 = 'v';
char letter4 = 'a';

System.out.print("The letters of the word ");
System.out.print(letter1);
System.out.print(letter2);
System.out.print(letter3);
System.out.println(letter4);
System.out.print("have Unicode values: ");
System.out.print((int)letter1 + ", ");
System.out.print((int)letter2 + ", ");
System.out.print((int)letter3 + " and ");
System.out.println((int)letter4);
```


4. Determine the exact output to the following code segment:

```
int sum1 = 10+5;
double sum2 = 10+5.0;
int quotient1 = 15/4;
double quotient2 = 15/4;
double quotient3 = 15/4.0;
int remainder1 = 15 % 4;
double remainder2 = 17.84 % 2.3;

System.out.println("10+5 is " + sum1);
System.out.println("10+5.0 is " + sum2);
System.out.println("15/4 is " + quotient1);
System.out.println("15/4 is " + quotient2);
System.out.println("15/4.0 is " + quotient3);
System.out.println("15 % 4 is " + remainder1);
System.out.println("17.84 % 2.3 is " + remainder2);
```

5. As discussed in chapter 1, integers are stored as binary values. The base 10 numbers 27 and 21 (using short ints) are stored as 0000000000011011 and 0000000000010101 respectively. Add these two binary numbers together and convert the sum back to base 10. Did you get 48?

<u>Precedence and Assignment Operators</u>

Using precedence rules, determine the correct answer to the following expressions:

1. 8 + 3 * 6 / 5 % 6 - 9

2. (8 * 3) / 9 + 2 * 5

3. (**double**) 9 / 4

4. (**int**) 17.5 / 3

5. 12 * 3 % 8 / 2

6. (12 * 3) % (8 / 2)

7. 17.5 / 3.75 + 2

8. 12.5 % 3

Translate each of the following statements into Java code.

9. Increment *number* by 10.

a. longer version

b. using assignment operator

10. Increment *count* by 1.

a. longer version

b. using increment operator

11. Multiply *base* by 2.

a. longer version

b. using assignment operator

12. What is the most important reason for using constants in a program?

String Class ("if-else" preview)

Find the output to the following code segments:

1.

```
String state = "Mississippi";
int len = state.length();
char ch = state.charAt(8);
String stub = "ssip";
int index1 = state.indexOf(stub);
int index2 = state.indexOf('s');
System.out.println("The length of " + state + " is " + len
  + " characters");
System.out.println("The character at index 8 is " + ch);
System.out.println("The beginning index of " + stub + " is "
  + index1);
System.out.println("The first index value of s is "+ index2);
```

2.

```
String state = "Washington";
String abbrev = state.substring(0,2);
System.out.println(abbrev.toUpperCase() + " ships "
  + state.substring(7) + "s of apples");
```

3.

```
String state = "California";
if (state.equals(state.toLowerCase()))
  System.out.println("I guess caps don't really matter.");
else
  System.out.println("Picky, picky!");
if (state.equalsIgnoreCase(state.toLowerCase()))
  System.out.println("That gives us some flexibility.");
else
  System.out.println("Are you sure?");
```

