Assignment\_2\_readingAndQuestions

Due date:

Course: FDU CSCI-3307 Java

Instructor: ilker Kiris

Student: **Paul Jerchaflie**

**DIRECTIONS**

1. **Copy** this file to **ass2-readingAndQuestions-yournameLastName** and work on it
2. Please read the below specified sections from our book

Java - How to Program, Early Objects”, Deitel & Deitel", latest edition (i.e Early Objects 10th Edition)

1. Write your answers to book questions properly labeled.
2. Write your answers to provided “Related questions” below them
3. For questions that are programming, follow the directions in below section

**SETUP and how to submit programming questions**

1. Create Java Project (highly suggesting to create it in STS) called ass-2-littlePrograms and put the java packages into a source directory called src
2. You can use <https://github.com/fdu-3307/example-1-operationsOn2numbers>, as how you should organize packages, what to have in .gitignore, README.md, …,
3. Use package and file names like following example. For example for **question** **2.6** in **chapter 2** that requires programming, create a package **com.mycompany.ass2.chapter2** , create a class called **Program\_2\_6**.
4. Have a README.md file at root (top) level of project explaining what is happening
5. Create a class diagram (highly suggesting to install ObjectAid eclipse plugin to STS and using that to create class diagram and an image of the class diagram and put them to directory called *design* under root level). Call that classDiagram , have all your program classes in it.
6. Use you github organization called **csci3307-ass-yournameLastname**. Create it if you have not already.
7. Create a github repository called **ass-2-littlePrograms-yournameLastname** in above organization
8. Create a local project directory based git repository of your project, commit your code (first add to index then commit) to local repository, then set above created github repository as remote repository for the project and finally push to remote github repository
9. **Email** ilker at ilkerkiris@gmail.com the link to your above created github repository. **Also** make sure to write that link in your copy of this file.

**Book’s Chapter 1 - Sections to read and book questions to answer**

1. Read sections 1.5, 1.8, 1.9.
2. Answer questions 1.2, 1.3, 1.6, 1.7, 1.8

1.2

1. the **JAVA** command
2. the **JAVAC** command
3. .java
4. .class
5. Java byte code

1.3

1. Information hiding
2. classes
3. Objected oriented programming
4. Using inheritance
5. UML
6. attributes

1.6

1. Edit, compile, load, verify, and execute
2. IDEs
3. JVM
4. Virtual Machine
5. JVM’s class loader
6. Bytecode verifier

1.7

In the first stage the java compiler compiles a .java file to a .class bytecode

In the second stage, the JVM executes the bytecode to run the program

1.8

**Object**- the specific instance of a watch, ex. Bob’s watch

**Attributes**- current hour, minute, second, length of wrist strap

**Behaviors**-tellTime, setTime

**Class**-the blueprint from which all watches of a specific model are made from, there are copies of the same watch for sale because they are made from the same blueprint

**Inheritance**- a wrist watch has many of the same behaviors and attributes as an alarm clock, but also has some attributes that are specific to it, like a wrist strap

**Modeling**- a wristwatch and its parents could be modeled using UML, the model would also highlight the behaviors and attributes of the watch

**Messages**- messages are implemented method calls, a message for a watch would be turning the notch on the side of the watch to change the time. Turning the watch sends a message to the other components of the watch (depending on whether it is a digital or mechanical watch determines whether the message is sent mechanically or digitally)

**Encapsulation**- The user of the watch does not have full access to the attributes of the watch. The user must change the time of the watch from the methods provided to him or her.

**Interface and information hiding**- The user of the watch does not have access to the inside of the watch. However, the user does not need access to the innards of the watch in order to use it properly. In fact, the key components of the watch that make it run are hidden from the user in order to prevent the user from causing problems

**Related questions (RQ)**

1. Who invented/crated Java?

James Gosling

1. Which company originally created the Java?

Sun Microsystems

1. What was original name of Java?

Oak

1. Which year Java was first released?

JDK 1.0 was released 1996

1. Which field were Java creators trying to provide solution to?

They were unsatisfied with their C++ APIs and decided to make their own opinionated language

1. Which company currently owns Java?

Oracle

1. What is the latest Java version currently?

Java 8

1. Which language does Android use?

Java

1. Roughly how many cell phones can run Java worldwide?

There are 100+ Million android phones worldwide

1. Roughly how many Java developers are there in the world?

9 Million

1. What does JVM stand for?

Java Virtual Machine

1. What do you need JVM for? What does JVM do?

JVM execute Java bytecode, this in part allows java to run on any operating system

1. What does JRE stand for?

Java Runtime environment

1. What do you need JRE for?

To run java applications

1. What is the relationship btw JVM and JRE?

JRE contains JVM

1. What does JDK stand for?

Java Development Kit

1. What do you need JDK for?

It contains the java compiler which is used to create .class files from .java files

1. What is the relationship between JRE and JDK?

JDK contains JRE

1. Can Java run on any machine? What is needed for a Java application to run on a computer?

Java can run on any machine and it needs JVM installed

1. What is a keyword? List some Java keyword.

Keywords are words reserved for specific functions, you cannot name a variable a keyword. Examples of keywords are: class, extends, final, super

1. Is Java case sensitive? What is the case for Java keywords?

Yes, all lowercase

1. How do you create a line comment in Java?

//

1. How do you create multi-line (block) comment in Java?

/\*

Create a block comment like this

\*/

1. What command line tool do you use to compile a Java file?

Javac

1. What command line tool do you use to run a Java byte (class) file?

java

1. What is the Java source file extension?

.java

1. What is the Java byte code file extension?

.class

1. What gets created when you do below at command line?

javac MyClass.java

MyClass.class is created

1. What does a class like MyClass needs to have for it be an entry point of a Java application?

A main method

1. What happens when you type below at command line?

Java MyClass

It will run the main method of MyClass

1. What is the output of following code?

public class TestMe1 {

public static void main(String[] args) {

System.out.println(“3.5 \* 4 / 2 – 2.5 is ”);

System.out.println(3.5 \* 4 / 2 – 2.5);

}

}

3.5 \* 4 / 2 – 2.5 is 4.5

1. Which of below option(s) is/are correct for a main method? Put OK after it.

public static void main(String[] args) { …..} OK

public static void main(String args[]) { …}OK

public static void main(String args) { ..}

public static void main() {...}

1. If a NoClassDefFoundError occurs when you run a program, what is the cause of it?

The class file needed to run a program is missing, it was there at compile time but it not available during runtime

1. If a NoSuchMethodError occurs when you run a program, what is the cause of it?

There is no main method or it is not properly implemented

1. What is a syntax error?

You get a syntax error when you don’t follow the “grammar” of java

Examples of syntax errors are missing semicolons or mismatched paranthesis

1. How do you know you have syntax error from command line?

Your source code will fail to compile and you will receive an error

1. How do you know you have syntax error in STS (eclipse)?

The editor will underline the error in red squiggly lines

1. What is a runtime error? Give an example.

A runtime error is any error that occurs while the program is running

Division by zero is an example of a runtime error

**Book’s Chapter 2 - Sections to read and book questions to answer**

1. Read sections 2.2, 2.5.1, 2.5.4, 2.5.6
2. Answer questions 2.1, 2.2, 2.3, 2.4, 2.5, 2.6 (this is a programming one), 2.7, 2.8, 2.9, 2.10, 2.11, 2.15 (programming), 2.16(programming), 2.17(programming), 2.18(programming, call it Program\_2\_18)

2.1

1. Left curly brace {, right curly brace }
2. If
3. //
4. Spaces, line breaks, and tabs
5. Keywords
6. Public static void main(String [] args)
7. System.out, System.out.println, System.out.print

2.2

1. False, comments cause the compiler to ignore what is after the //
2. True
3. False, Java is case sensitive
4. True
5. False, arithmetic operators follow PEMDAS precedence

2.3

1. int c,thisIsAVariable, q76354;
2. System.out.println(“Enter an Integer”);
3. Scanner input =new Scanner(System.in);

int inp=input.nextInt();

1. System.out.println(“This is a Java program”);
2. System.out.printf(“This is a Java%sprogram%s”);
3. if (number!=7){ System.out.println(“The variable number is not equal to 7”);}

2.4

Identify and correct the errors in each of the following statements:

a) if (c < 7);

System.out.println("c is less than 7");

Remove semicolon

b) if (c => 7)

System.out.println("c is equal to or greater than 7");  
switch order of => to >=

2.5

1. //This program will calculate the product of three integers
2. Scanner input=new Scanner(System.in);
3. int x,y,z,result;
4. System.out.println(“Enter an integer”);
5. x=input.nextInt();
6. System.out.println(“Enter another integer”);
7. y=input.nextInt();
8. System.out.println(“Enter another integer”);
9. z=input.nextInt();
10. result=x\*y\*z;
11. System.out.printf(“Product is%d”,result);

2.7

1. Comments
2. if statement
3. assignment
4. division, remainder
5. innermost
6. variable

2.8

1. System.out.print(“Enter and integer: ”);
2. a=b\*c;
3. //This program perfoms a samlple payroll calculation

2.9

1. False, they are evaluated in order of precedence that algebra follows and are therefore not always left to right
2. True
3. False, division, multiplication, and remainders are calculated before addition and subtraction
4. False, h22 is a valid variable name

2.10

1. x = 2
2. Value of 2 + 2 is 4
3. x =
4. 5 = 5

2.11

Which of the following Java statements contain variables whose values are modified?

a,d

**Related questions (RQ)**

1. What is equivalent of below? Tip; following order of execution per operator precedence, wrap the sections that operate first in (..)

a – b + c –d

((a-b)+c)-d

1. What is equivalent of below? What is its final value?

1 + 2 \* 3 > 4 \* (5 + 6) && (7 – 8 > 9)

(1 + (2 \* 3)) > (4 \* (5 + 6)) && ((7 – 8) > 9)

False

1. What is equivalent of below?

a= b += c = 5

c=5

b=b+c

a=b

1. What is value of below statements?

true || true && false False

true && true || false True

!true False

true ^ false True

1. What is “!” operator?

not

1. What is “^” operator?

XOR (exclusive or)

1. Write a program (call it Program\_RQ\_2\_1), that displays the result of
   1. x 2.5 – 1.3 x 3

-----------------------

31.5 – 1.5

**Book’s Chapter 3 - Sections to read and book questions to answer**

1. Quick scan read sections 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.5.1, 3.5.2
2. Read and understand the code in figures 3.8, 3.9
3. Answer questions 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.10, 3.11(programming)

3.1

1. public
2. class
3. new
4. type and name
5. default package
6. float and double
7. double precision
8. nextDouble()
9. access specifier
10. void
11. getLine()
12. java.lang
13. import declartation
14. floating point
15. single
16. %d
17. Primitive and reference

3.2

1. False, the first letter is lowercase
2. True
3. True
4. False, primitives do not have methods
5. False, they are restricted within the scope of the method
6. True
7. False, a primitive local variable can be declared but not initialized
8. True
9. True
10. True
11. False, they are double by default

3.3

Local variables are declared statically, which means the memory is reserved for them at compile time. Instance variables are declared dynamically which means the memory that they occupy is reserved during run-time.

3.4

A parameter is used to specify what types can be passed to a method. A parameter specifies the type that can be passed to a method while an argument is the actual value and type being passed to a method.

3.5

The new keyword is used to create instance variables. When the new keyword is used it dynamically allocates a place in memory for the variable all during run time.

3.6

The default constructor is the constructor that is used by default if one is not specified. When the default constructor is called the instance variables of a class are not initialized. The only way to initialize them would be to call a setter method.

3.7

Instance variables are used to represent the attributes of a class. Each instance of a class each has its own instance variables specific to that object.

3.10

A class might have a get and set method for an instance variable because this is more secure than giving full access to the instance variable by setting it to public. By using setters and getters, the instance variable is encapsulated and the use of the instance variable is restricted to an intended purpose.

**Related questions (RQ)**

1. What is “import” keyword for? Let’s say there is a class called MyClass in a package com.mycompany.app1.comp1, how can you import it in another class?

Import is used to include classes from a different package, it allows one to use code from another source

import com.mycompany.app1.comp1

1. What is “static” keyword for?

Static allows you to call a function without needing an instance of that class

1. What is “final” keyword for?

Final is used to denote a variable as a constant

1. What are 4 access modifiers in Java?’

Public

Private

Protected

package

1. What does below in a class MyClass mean?

public static final String DEFAULT\_NAME = “ilker”;

this creates a constant class string

1. What do 4 java access modifiers mean?

Public –everyone has access

Private – access only from within the class

Protected – children have access as well

Package – within its package

1. Why is main method “static void main”?

It is static because you don’t want to create an instance to call the function on

Void because you don’t want to return any value

The main method must be called main to denote it as the main method

1. Give 3 “primitive data type” and their wrapper Objects?

int Integer

double Double

boolean Boolean

1. What is the difference between boolean and Boolean?

boolean is the primitive data type, Boolean is the wrapper object of boolean

**Book’s Chapter 4 - Sections to read and book questions to answer**

1. Quick scan read sections 4.7, 4.8
2. Read and understand the code in figures 4.12, 4.13, 4.14
3. Write programs for code that is in figures; (4.4 + 4.5 combined), 4.8, 4.10, 4.15
4. Answer questions 4.8, 4.15, 4.16, 4.20 (programming), 4.24 (programming), 4.25, 4.26

4.8

Identify and correct the errors in each of the following sets of code:

a)

while (c <= 5)

{

product \*= c;

++c;

Has a missing right curly brace

b)

if (gender == 1)

System.out.println("Woman");

else;

System.out.println("Man");  
unnecessary semi colon to the right of else

4.15

a)

if (age >= 65);

System.out.println("Age is greater than or equal to 65");

else

System.out.println("Age is less than 65)";

Semi colon

b)

int x = 1, total;

while (x <= 10)

{

total += x;

++x;

}

Total is never initialized, you cannot do += on a variable that’s not initialized

c)

while (x <= 100)

total += x;

++x;

This loop is missing curly braces, because of this, the loop will run forever

d)

while (y > 0)

{

System.out.println(y);

++y;  
There is a missing right curly brace: }

4.16

1

4

9

16

25

36

49

64

81

100

Total is 385

4.25

\*\*\*\*

++++++++

\*\*\*\*

++++++++

\*\*\*\*

++++++++

\*\*\*\*

++++++++

\*\*\*\*

++++++++

4.26

>>>>>>>>>>

<<<<<<<<<<

>>>>>>>>>>

<<<<<<<<<<

>>>>>>>>>>

<<<<<<<<<<

>>>>>>>>>>

<<<<<<<<<<

>>>>>>>>>>

<<<<<<<<<<

**Related questions (RQ)**

1. What is wrong with following while statement?

int sum = 0, z = 0;

while (z <= 0 ) {

sum += z;

System.out.println(“z=” + z + “, sum=” + sum);

}

The value of sum never actually increases, the loop will go on forever because of this.

**Book’s Chapter 5 - Sections to read and book questions to answer**

1. Quick scan read top part of sections 5.4, 5.5, 5.6, 5.8
2. Read and understand the code in figures 5.6, 5.11 + 5.12
3. Write programs for code that is in figures; 5.1, 5.2,5.9,5.13
   1. Answer questions 5.2, 5.4, 5.7, 5.8, 5.9, 5.18 (programming based on Fig 5.6), (programming based on 5.11 + 5.12, make sure to have and if needed update caller/test code from 5.12)

5.2

1. False it is optional
2. False, it is optional
3. False that would only be the case with||
4. True
5. True
6. False there is so such functionality
7. true

5.4

a)

i = 1;

while (i <= 10);

++i;

}

Unnecessary semicolon

b)

for (k = 0.1; k != 1.0; k += 0.1)

System.out.println(k);

Need to add double to left of first k to specify type

c)

switch (n)

{

case 1:

System.out.println("The number is 1");

case 2:

System.out.println("The number is 2");

break;

default:

System.out.println("The number is not 1 or 2");

break;

}

There should be a break after the first print statement, otherwise it will print both “the number is 1” and “the number is 2” if the number is 1.

d)

The following code should print the values 1 to 10:

n = 1;

while (n < 10)

System.out.println(n++);

The code will only print the values from 1 to 9, either change boolean condition to n<11 or n<=10

5.7

If you want your loop to execute at least once, use a do while loop

This is useful if you are doing input validation. For example:

boolean valid=false;

do {

…

//{get some input into a var named input}

if (input==1){

doSomething();

valid=true;

}

...

//have some other cases

} while (!valid)

5.8

Break statements will terminate a loop and it will not continue to run, continue will just end the current iteration of the loop but the loop will continue to iterate normal afterwards

They are both similar in that they prevent the rest of the current iteration from exectuing

5.9

a)

For (i = 100, i >= 1, i++)

System.out.println(i);

This loop will run forever, the i++ should be changed to i--

b)

The following code should print whether integer value is odd or even:

switch (value % 2)

{

case 0:

System.out.println("Even integer");

case 1:

System.out.println("Odd integer");

}

After each print statement there should be a break statement, otherwise when the number is even both print statements will execute.

c)

The following code should output the odd integers from 19 to 1:

for (i = 19; i >= 1; i += 2)

System.out.println(i);

This loop will execute forever printing odd numbers from 19 and up

To fix this change i+=2 to i-=2

**Related questions (RQ)**

1. What is the difference between while(someCondition) {blabla} and do {blabla} while(someCondition)?

While will only execute if the condition is true

Do while will always execute at least once

1. What is the difference between “break” and “continue” in a loop?

Break will completely terminate the loop

Continue will just end the current iteration of the loop and will continue to iterate

1. What data types one can have as “key” in a switch statement? (like switch(key) {….}

byte, short, char, int, ( and their object wrappers) ,String, and enumerated types

1. What is the meaning of “case” in a switch statement?

Case acts somewhat like an if statement, when calling the switch statement you specify which variable you will be checking. In the body of the switch statement you can put case someValue: and if the argument passed to the switch statement is equal to someValue, the code after the case clause will execute.

1. What is the meaning of “break” in a switch statement? What happens when “break” statement executes in a switch statement?

Break defines the end of the body of code that should execute when a case statement is true.

1. What is the meaning of “default” in a switch statement?

Default acts very similar to else in an if statement, if none of the cases were true, the default section will run.

1. Is “break” optional in a switch statement?

Yes

1. Is “default” optional in a switch statement?

Yes but it should generally be used.

1. What happens if you don’t have “break” after a case statement in a switch statement?

If it is omitted, all of the code in the case statements will run until a break is reached

1. What happens if you don’t have “default” in a switch statement?

If the default section is omitted and would be called, nothing will happen

1. Is below if statement correct? If not what is wrong with it? If not what is the correct way to do that?

int dayOfMonth = 12;

if(1 <= dayOfMonth <= 31) {…..bla bla…}

no do if (1<=dayOfMonth&&dayOfMonth<=31) instead