**CSC 130 – Exam 1**

**Multiple choice questions – 5 points each**

* 1. Java is an object-oriented programming language. An object-oriented language

1. Uses structured programming.
2. Views a program as consisting of objects which communicate through interactions.
3. Functionally breaks down problems into smaller, more manageable problems.
4. All of the above.

The answer is D

* 1. In Java, source code is compiled into object code called Byte-code.

1. Bit-code
2. Class code
3. Method code
4. Byte-code

The answer is D

* 1. Identify the invalid Java identifier.

1. 1Week variable name cannot start with a number
2. Week1
3. amountDue
4. amount\_due

The answer is A

* 1. What is the value of 7.52e-5? => 7.52 x 10^-5 =>0.0000752

1. 752000.0
2. 0.0000752
3. 0.000752
4. 0.00752

The answer is B

* 1. What is the Java expression for 4a2 + 2b \* c?

1. (4 \* a) + (2 \* b) \* c
2. (4 \* a \* a) + ((2 \* b) \* c) multiplication happens before addition
3. ((4 \* a \* a) + (2 \* b)) \* c
4. (4 + a \* a) + ((2 + b) \* c)

The answer is B

* 1. Which operator is used to concatenate two strings?

1. +
2. –
3. \*
4. /

The answer is A

* 1. What is the value of the variable c in the statements that follow?

String phrase = "Make hay while the sun is shining.";

**char** c = phrase.charAt(10);

1. w
2. h “M” is at index 0
3. i
4. None of the above

The answer is B

* 1. In Java, a block comment is delimited by:

1. \*/ /\*
2. /\* /\*
3. /\* \*/
4. \*/ \*/

The answer is C

* 1. The syntax that declares a Java named constant named SALES\_TAX is:

1. double SALES\_TAX = 7.50;
2. public double SALES\_TAX = 7.50;
3. public static double SALES\_TAX = 7.50;
4. public static final double SALES\_TAX = 7.50; key word final is required

The answer is D

* 1. The escape sequence the represents the new-line character is:

1. \r escape sequence for carriage return
2. \t escape sequence for tab
3. \n escape sequence for new line
4. \\ escape sequence for back slash

The answer is C

* 1. Which statement or group of statements produces the output: Java programming is fun!

1. System.out.print(Java programming);

System.out.print(is fun!);

1. System.out.println(Java programming is fun!);
2. System.out.println(“Java programming”);

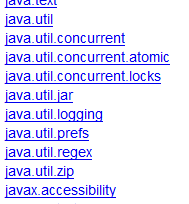
System.out.println(“ is fun!”); this will print on 2 lines

1. System.out.print(“Java programming”)

System.out.println(“ is fun!”); assuming there is a blank space between “ and is

The answer is D

* 1. The class NumberFormat allows you to specify a constant representing which country’s currency format should be used. To use this constant you must import:

1. [](http://docs.oracle.com/javase/1.5.0/docs/api/)java.util.Locale there no such packages:
2. java.util.Currency
3. java.util.Properties
4. None of the above.

The answer is D

* 1. What Java package includes the class Scanner?

1. awt
2. swing
3. io
4. util

The answer is D

* 1. A compound statement is enclosed between:

1. [ ]
2. { }
3. ( )
4. < >

The answer is B

* 1. A multi-way if-else statement

1. allows you to choose one course of action. No, it allows for a different course based on which Boolean condition is true.
2. always executes the else statement.
3. allows you to choose among alternative courses of action. But only one of them. A multi-branch if-else statement never performs more than one action.
4. executes all Boolean conditions that evaluate to true.

The answer is C

* 1. To compare two strings lexicographically the String method compareTo should be used.

1. equals
2. equalsIgnoreCase
3. compareTo
4. ==

The answer is C

* 1. The OR operator in Java is represented by:

1. ! this is NOT
2. && this is AND
3. | |
4. None of the above

The answer is C

* 1. The dot operator has the highest precedence.

1. \*
2. dot
3. +=
4. Decrement after dot or left-associative

The answer is B

* 1. The negation operator in Java is represented by:

1. !
2. && this is AND
3. | | this is OR
4. None of the above
   1. A mixture of programming language and human language is known as:
5. Algorithms can be written in pseudocode or a programing language
6. Recipes
7. Directions
8. Pseudocode

**Short Answers Questions (10 points each)**

* 1. What are the values of the variables a, b, c, and d after the execution of the following expressions?

**int** a = 3;

**int** b = 12;

**int** c = 6;

**int** d = 1;

d = d \* a; 🡺 d = 1\*3 🡺 d =3

c = c + 2 \* a; 🡺 c = c + 6 🡺 c = 12

d = d - b / c; 🡺 d = 3 – 12 / 12 🡺 d = 3 – 1 🡺 d = 2

c = c \* b % c; 🡺 c = 12 \* 12 % 12 🡺 c = 0 these are in the same order of precedence so just left to right

b = b / 2; 🡺 b = 12 / 2 🡺 b = 6

* 1. What is the output produced by the following lines of code?

**int** value1 = 3;

**int** value2 = 4;

**int** result = 0;

result = value1++ \* value2--; 🡺 12 now value1 is 4 and value2 is 3

System.out.println("Post increment/decrement: " + result);

Post increment/decrement: 12

result = ++value1 \* --value2; now value1 is 5 and value2 is 2 so result is 10

System.out.println("Pre increment/decrement: " + result);

Pre increment/decrement: 10

|  |
| --- |
| Post increment/decrement: 12  Pre increment/decrement: 10 |

* 1. Write a Java statement to display your name in the console window.

System.out.println(“Casey Carnnia”);

System.out.print(“Casey Carnnia”);

* 1. Why is echoing user input a good programming practice?

To help the user know that the value that the application is working with is exactly the value she entered. It helps in debugging to.

* 1. Write a complete Java console application that prompts the user for two numbers, multiplies the numbers, and then displays the result to the user.

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.10.2012  Scope: Write a complete Java console application that prompts the user for two numbers,  multiplies the numbers, and then displays the result to the user.  Solution: Import Scanner class from java.util package  Welcome the user  prompt for input one INPUT\_NUM1 type = int  prompt for input one INPUT\_NUM2 type = int  store the input in constants INPUT\_NUM1 INPUT\_NUM2 these values will not change  declare variable to hold the multiplication outputAnswer type = int  outputAnswer = INPUT\_NUM1 \* INPUT\_NUM2  output: "INPUT\_NUM1 times INPUT\_NUM2 equals outputAnswer"      \*/    // declare class  **public** **class** multiplyNum1Num2{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("I can multiply 2 numbers for you.");  // get ready to read the user data  Scanner keyboard = **new** Scanner(System.*in*);  // prompt the user for INPUT\_NUM1  // store the user input  System.*out*.print("Please enter a number: ");  **final** **int** INPUT\_NUM1 = keyboard.nextInt();    // prompt the user for INPUT\_NUM2  // store the user input  System.*out*.print("Please enter another number: ");  **final** **int** INPUT\_NUM2 = keyboard.nextInt();    // declare variable for answer  **int** outputAnswer = 0;    // do the calculation  outputAnswer = INPUT\_NUM1 \* INPUT\_NUM2;    //output to the console  System.*out*.println(INPUT\_NUM1 + " times " + INPUT\_NUM2 + " equals " + outputAnswer);    }//main END   1. }//class END |
|  |

* 1. What output will be produced by the following code?

**public** **class** SelectionStatements

{

**public** **static** **void** main(String[] args)

{

**int** number = 24;

**if**(number % 2 == 0)

System.out.print("The condition evaluated to true!");

**else**

System.out.print("The condition evaluated to false!");

}

}

The condition evaluated to true! Because 24/2 has no remainder

* 1. Write a multi-way if-else statement that evaluates a persons weight on the following criteria: A weight less than 115 pounds, output: Eat 5 banana splits! A weight between 116 pounds and 130 pounds, output: Eat a banana split! A weight between 131 pounds and 200 pounds, output: Perfect! A weight greater than 200 pounds, output: Plenty of banana splits have been consumed!

Based on the requirement the application produces no results if weight = 115

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.10.2012  Scope: Write a multi-way if-else statement that evaluates a persons weight on the following criteria:  A weight less than 115 pounds, output: Eat 5 banana splits!  A weight between 116 pounds and 130 pounds, output: Eat a banana split!  A weight between 131 pounds and 200 pounds, output: Perfect!  A weight greater than 200 pounds, output: Plenty of banana splits have been consumed!  Solution: Import Scanner class from java.util package  Welcome the user  prompt for INPUT\_WEIGHT type = int  store the input in constants INPUT\_WEIGHT values will not change  declare a variable msg type = String  if INPUT\_WEIGHT < 115  msg = "Eat 5 banana splits!"  else if INPUT\_WEIGHT > 115 AND INPUT\_WEIGHT <= 130  msg = "Eat a banana splits!"  else if INPUT\_WEIGHT > 130 AND INPUT\_WEIGHT <= 200  msg = "Perfect!"  else if INPUT\_WEIGHT > 200  msg = "Plenty of banana splits have been consumed!"      output: "msg"      \*/    // declare class  **public** **class** bananaSplit{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("If you tell me your weight I'll tell you how many banana splits you should eat.");  // get ready to read the user data  Scanner keyboard = **new** Scanner(System.*in*);  // prompt the user for INPUT\_WEIGHT  // store the user input  System.*out*.print("Please enter your weight: ");  **final** **int** INPUT\_WEIGHT = keyboard.nextInt();      // declare variable for answer  String msg = "";    // do the calculation  **if** (INPUT\_WEIGHT < 115){  msg = "Eat 5 banana splits!";  }  **else** **if** (INPUT\_WEIGHT > 115 && INPUT\_WEIGHT <= 130){  msg = "Eat a banana splits!";  }  **else** **if** (INPUT\_WEIGHT > 130 && INPUT\_WEIGHT <= 200){  msg = "Perfect!";  }  **else** **if** (INPUT\_WEIGHT > 200){  msg = "Plenty of banana splits have been consumed!";  }//if END    //output to the console  System.*out*.println(msg);    }//main END  }//class END |
|  |

* 1. Name and describe three types of branching mechanisms and give an example of each.

Three types of branching mechanisms include the if-else statement, the multi-way if-else statement, and the switch statement. An if-else statement chooses between two alternative statements based on the value of a Boolean expression. If you only need to perform an action if a certain condition is true, then the else part of an if-else statement can be omitted. A multi-way if-else statement allows several conditions to be evaluated. The first condition that evaluates to true, is the branch that is executed. The switch statement is another multi-selection branching mechanism. A switch evaluates a controlling expression and executes the statements in the matching case label.

The answer is already provided!

Write an if-else statement to compute the amount of shipping due on an online sale. If the cost of the purchase is less than $20, the shipping cost is $5.99. If the cost of the purchase over $20 and at most $65, the shipping cost is $10.99. If the cost of the purchase is over $65, the shipping cost is $15.99.

Based on requirements the application does not charge shipping when cost of purchase is == 20

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.10.2012  Scope: Write an if-else statement to compute the amount of shipping due on an online sale.  If the cost of the purchase is less than $20, the shipping cost is $5.99.  If the cost of the purchase over $20 and at most $65, the shipping cost is $10.99.  If the cost of the purchase is over $65, the shipping cost is $15.99.  Solution: Import Scanner class from java.util package  Welcome the user  prompt for INPUT\_PRICE type = double  store the input in constants INPUT\_PRICE values will not change  declare a variable shipping type = double  if INPUT\_PRICE < 20  shipping = 5.99  else if INPUT\_PRICE > 20 AND INPUT\_PRICE <= 65  shipping = 10.99  else if INPUT\_PRICE > 65  shipping = 15.99    output: "Cost of shipping is $ shipping"      \*/    // declare class  **public** **class** shippingDue{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("I can calculate the shipping charges based on the purchase price.");  // get ready to read the user data  Scanner keyboard = **new** Scanner(System.*in*);  // prompt the user for INPUT\_PRICE  // store the user input  System.*out*.print("Please enter price of your purchases: ");  **final** **double** INPUT\_PRICE = keyboard.nextDouble();  // declare variable for answer  **double** shipping = 0;    // do the calculation  **if** (INPUT\_PRICE < 20){  shipping = 5.99;  }  **else** **if** (INPUT\_PRICE > 20 && INPUT\_PRICE <= 65){  shipping = 10.99;  }  **else** **if** (INPUT\_PRICE > 65){  shipping = 15.99;  }    //output to the console  System.*out*.printf("Cost of shipping is $%4.2f ",shipping);    }//main END  }//class END |
|  |

* 1. What is short-circuit evaluation of Boolean expressions?

Java evaluates the first sub-expression in expressions involving || or &&, and if that is enough information to learn the value of the whole expression, it does not evaluate the subsequent sub-expressions. This is called short-circuit evaluation or lazy evaluation.