|  |  |
| --- | --- |
| **Question 1:** | Briefly explain what is meant by the *syntax* and the *semantics* of a programming language. Give an example to illustrate the difference between a syntax error and a semantics error.  **Syntax errors** : These are invalid code the compiler doesn't understand: int a = 5  Compiler message:  Example.java:20: ';' expected  **Semantic errors** : These are valid code the compiler understands, but they do not what you, the programmer, intended.  int i;  i++; // the variable i is not initialized |
| **Question 2:** | What does the computer do when it executes a variable declaration statement. Give an example.  int x;  which creates the variable x. When the computer executes a variable declaration, it creates the box in memory and associates a name (in this case, x) with that box. Later in the program, that variable can be referred to by name. |
| **Question 3:** | What is a *type,* as this term relates to programming?  For example, to say that a variable is of type int says that integer values in a certain range can be stored in that variable. |
| **Question 4:** | One of the primitive types in Java is *boolean.* What is the boolean type? Where are boolean values used? What are its possible values?  The only values of type Boolean are TRUE and FALSE. They are used in conditions like while loops and if statements. |
| **Question 5:** | Give the meaning of each of the following Java operators:  a) ++ int plus 1  b) && represents the word AND. X>0 && X<4  c) != not equel to. 1 != 2 |

|  |  |
| --- | --- |
| **Question 6:** | Explain what is meant by an *assignment statement,* and give an example. What are assignment statements used for?  An assignment statement computes a value and stores that value in a variable.  x = 8; // Assign a constant value to the variable, x.  newRow = row; // Copy the value from the variable, row,  // into the variable,newRow.  ans = 8\*x + ; // Compute the value of the expression  // 17\*x + 42, and store that value in ans.  An assignment statement is used to change the value of a variable as the program is running. |
| **Question 7:** | What is meant by *precedence* of operators?  For example, \* has higher precedence than +, so the expression 3+5\*7 is evaluated as if it were written 3+(5\*7). |
| **Question 8:** | What is a *literal*?  For example, 'A' is a literal that represents the value A, of type char, and 17L is a literal that represents the number 17 as a value of type long. A literal is a way of writing a value. |
| **Question 9:** | In Java, classes have two fundamentally different purposes. What are they?  A class can be used to group together variables and subroutines that are contained in the class. The second possible purpose of a class is to describe and create objects. The class specifies what variables and subroutines are contained in those objects. |
| **Question 10:** | What is the difference between the statement "x = TextIO.getDouble();" and the statement "x = TextIO.getlnDouble();"  Either statements will read a real number input by the user, and store that number in the variable, x. The difference is that in the second statement, using getlnDouble, after reading the value, the computer will continue reading characters from input up to and including the next carriage return. |

|  |  |
| --- | --- |
| **Question 11:** | Explain why the value of the expression 2 + 3 + "test" is the string "5test" while the value of the expression "test" + 2 + 3 is the string "test23". What is the value of "test" + 2 \* 3 ?  2 + 3 + "test" is interpreted as (2 + 3) + "test", so 2 and 3 are added together, giving 5, and then the 5 is concatenated onto the string "test". On the other hand, "test" + 2 + 3 is interpreted as ("test" + 2) + 3, so the 2 is first concatenated onto the "test", giving "test2", and then the 3 is concatenated onto that. Since \* has higher precedence, this expression is interpreted as "test" + (2 \* 3), which evaluates to "test6". |
| **Question 12:** | Integrated Development Environments such as Eclipse often use syntax coloring, which assigns various colors to the characters in a program to reflect the syntax of the language. A student notices that Eclipse colors the word *String* differently from int, double, and boolean. The student asks why *String* should be a different color, since all these words are names of types. What's the answer to the student's question?  *String* is the name of a **class**, which int, double, and boolean are primitive types. Eclipse colors all class names in the same way that it does *String*, and it uses a different color for the primitive types. |