|  |
| --- |
| **Question 1 Andrew Pettus ITSE-1302 T/Th 4pm** |
| Assume that the following code is in your program  int myNumbers[5] = { 16, 2, 77, 40, 12071 };  What would the following code output:  cout << myNumbers[1]; |
| 2 |
| **Question 2** |
| What is the output of the following code example:  for(int i = 5; i < 10; i++)      {          cout << i << endl;      } |
| 5  6  7  8  9 |
| **Question 3** |
| What is the output of the following code example:  int counter = 10;      while(counter < 10)      {          cout << "Hello world!" << endl;      }  cout << "End of Program" << endl; |
| End of Program |
| **Question 4** |
| What is the output of the following code example:  int myNumber = 10;      if( (myNumber % 2 == 0) && (myNumber % 5 == 0) )      {          cout << "Hello world!" << endl;      }      else      {          cout << "Hello student!" << endl;      } |
| Hello world! |
| **Question 5** |
| What would the output of the following code be?  string student[5] = { "Sam", "Fred", "Sally", "Kevin", "Mary"};      double studentGrade[5] = {98.2, 78.5, 96.3, 89.1, 88.0};        cout << "Student: " << student[3] << " Grade: " << studentGrade[3] << endl; |
| Student: Kevin Grade: 89.1 |
| **Question 6** |
| How many times will the following code output Hello World  int myCounter = 5;      int myNumber = 15;      do      {          cout << "Hello World!" << endl;      } while (myNumber < 10); |
| One time |
| **Question 7** |
| The expression  static\_cast<int>(19.9)  evaluates to \_\_\_\_. |
| 19 |
| **Question 8** |
| Which of the following struct definitions is correct in C++?  A:  struct studentType  {    int ID;  };  B:  int struct studentType  {    int ID;  } |
| A:  struct studentType  {    int ID;  }; |
| **Question 9** |
| All components of an array are of the same data type.  **True or False** |
| True |
| **Question 10** |
| Arrays can be passed as parameters to a function by value, but it is faster to pass them by reference  **True or False** |
| False |
| **Question 11** |
| Assume you have the following declaration char nameList[100];. Which of the following ranges is valid for the  index of the array nameList?  A. 0 through 99  B. 0 through 100  C. 1 through 100  D. 1 through 101 |
| 1. 0 through 99 |
| **Question 12** |
| The data type of a variable in a return statement must match the function type.  **True or False** |
| True |
|  |
| **Question 13** |
| The expression (x >= 0 && x <= 100) evaluates to false if either x < 0 or x >= 100.  **True or False** |
| False |
| **Question 14** |
| What does <= mean? |
| Less than or equal to |
| **Question 15** |
| What will the following expression output?  cout << pow(2.0, pow(3.0, 1.0)) << endl; |
| 8 |
| **Question 16** |
| What is the output of the following  cout statement  int x = 35;  int y = 45;  int z;  if (x > y)    z = x + y;  else    z = y – x;  cout << x << " " << y << " " << z << endl; |
| 35 45 10 |
| **Question 17** |
| Given the following prototype, which selection below would be the correct usage of the function?  int test(float, char);  A. cout << test(12, &);  B. cout << test("12.0", "&");  C. int u = test(5.0, '\*');  D. cout << test("12", "&");  int u = test(5.0, '\*'); |
| **Question 18** |
| Given the following function:  int strange(int x, int y)  {    if (x > y)        return x + y;    else        return x – y;  }  What is the output of the following statement  cout << strange(4, 5) << endl;  -1 |
| **Question 19** |
| Briefly explain "Call by Value" and "Call by Reference" in you own words  Call by Value is when a function takes a copy of the value from a memory location for in scope / in-function  processing, once that number has been processed into a new number, unless passed back to a memory location,  the new number will be gone on the next function iteration. The original value is not modified in memory.   Call by Reference is when the memory location of the value is addressed specifically, then whatever processing  happens to the value is kept in the memory location where it originally resides. This allows for more efficient  access and changing of variables. The original value is modified. Pointers involved. |
| **Question 20** |
| What is the output of the following code  number = 1;  while (number < 5)  {    number++;    cout << number << " ";  }  cout << endl;  'number' was not declared in this scope, but if it was, this would output: **2 3 4 5** |
| **Question 21** |
| What is the output of the following C++ code?  count = 1;  num = 25;  while (count < 25)  {    num = num - 1;    count++;  }  cout << count << " " << num << endl;  ‘count’ and ‘num’ are not declared in this scope, but if they were, this would output: **25 1** |
| **Question 22** |
| What is the output of the following code  number = 1;  while (number < 5)  {    number++;    cout << number << " ";  }  cout << endl;  'number' was not declared in this scope, but if it was, this would output: **2 3 4 5**  **(same as Question 20?)** |
| **Question 23** |
| In C++ the identify the following operators:  && Logical AND operator  ||  Logical OR operator  % Modulo operator: produces remainder of integer division  !  Negation Boolean operator: (!true) = false; (!false) = true; (!=) = ‘is not equal to’ |
| **Question 24** |
| There is no right answer - the response is based on your opinion  This class was:   1. Too hard 2. **About right** 3. Too easy |
| **Question 25** |
| There is no right answer - the response is based on your opinion   1. What did you find most helpful in getting to the end of the class   Open book / Open note / Open code tests. Having references available to use while learning about HOW to implement the code was crucial for me. I would have not been able to memorize all this  information and be successful without resources.  I took Dr. Roark’s adage to heart, “Keep your old code open while you write your new code”.  Things came together pretty seamlessly with that logic.   1. If you were to change anything in the class - what would it be?   I would have preferred longer in-class lectures. I feel like I could have learned more information in  a single day, but that’s just me. I’ll probably eat these words during the next class.  I would have liked more lecture in correlation to the book, as it seemed like a handful of some  items found on the tests were not discussed in class. I feel like if it’s on the test, it should be brought  up in class at least once. Most items were discussed in class, but a couple test questions over the various  chapters had me trying to remember if we covered it.  Lastly I would have liked the labs to be a bit more challenging. Some of them I got done fairly quickly.  Again, that might be just me!  Dr. Roark is an amazing professor and I’m lucky to be his student learning software development. This class was  extremely informative and has launched me into the world of C++ and OOP development. I’m stoked for round 2. |
|  |