Section 1: Math & Physics

1. Given the following two linear equations, what is the point at which they intersect?

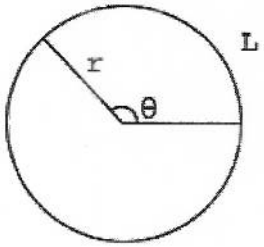
```
f(x) = 2x + 7
f(x) = -3x -2
```

a) They never intersect.
b) (-2.9, 6.4)
c) (6.9, -8.2)
d) (-1.8, 3.4)

2. Given the quadratic equation $f(x) = 3x^2 + 18x - 11$, at what point is the slope 0?

a) (100, 22)
b) (-3, -38)
c) (18, -11)
d) the slope is never 0

3. In the following diagram, $r = 10$ and $\theta = 0.72\pi$ rad. What is the approximate length of arc L?

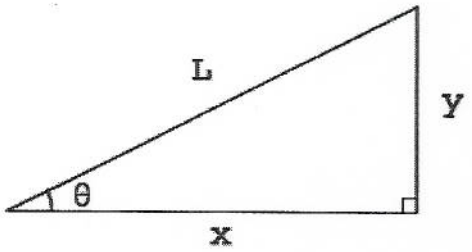


a) 25.39
b) 22.62
c) 6.28
d) 19.73

4. Given a right triangle with sides a, b, and c, where c is the hypotenuse, if $a = 7$ and $c = 10$, what is the approximate length of b?

a) 7.14
b) 6.58
c) 8.66
d) 9.48

5. In the following diagram, L = 8 and $\theta = 0.15\pi$ rad. What are the approximate values of x and y?



a) 6.22, 3.89
b) 7.13, 3.63
c) 8.36, 6.71
d) 6.07, 4.82

6. If vector **v** = [7, 10, 8] is multiplied by scalar m = 4, what is the resultant vector?

a) [36, 94, 86]
b) [10, 91, 57]
c) [63, 83, 64]
d) [28, 40, 32]

7. If vector **a** = [4, -3, 9] is added to vector **b** = [3, 2, -7], what is the resultant vector?

a) [7, -1, 2]
b) [2, -3, 0]
c) [3, -6, 4]
d) [7, -2, 5]

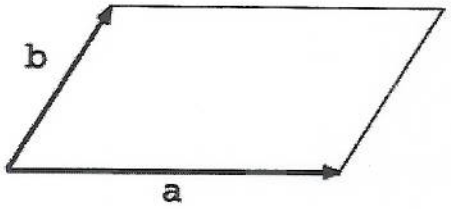
8. If you normalize vector **v** = [-7, 4, -8], what is the approximate resultant vector?

a) [-0.29, 0.54, -0.67]
b) [-0.01, 0.72, -0.08]
c) [-0.15, 0.49, -0.82]
d) [-0.61, 0.35, -0.70]

9. If vector **a** = [3, 5, 9] and vector **b** = [-9, 2, 0], what is the approximate angle between them (in radians)?

a) 0.09
b) 2.12
c) 0.86
d) 1.74

10. In the following diagram, **a** = [9, 1, 2] and **b** = [1, 3, -2]. What is the approximate normal of the plane described by the two vectors?



a) [-0.55, 0.36, 0.09]
b) [0.41, 0.30, -0.72]
c) [-0.33, -0.36, 0.95]
d) [-0.24, 0.59, 0.77]

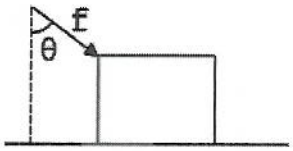
11. If an object is currently at 7m and is moving with a velocity of 3m/s, what point will it be at in 9 seconds?

a) 98m
b) 34m
c) 7m
d) 18m

12. If an object's initial position is 9m, has an initial velocity of 6m/s, and is experiencing a constant acceleration of $5m/s^2$, what position will it be at in 4 seconds?

a) 80m
b) 96m
c) 29m
d) 73m

13. In the following diagram, a force is being applied obliquely to a box that is resting on a level floor. If **f** = 9N and $\theta = 0.93$rad, approximately how much force is actually being applied horizontally to slide the box along the floor?

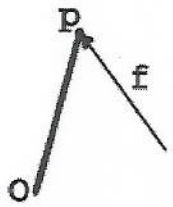


a) 1.38N
b) 7.21N
c) 8.08N
d) 4.69N

14. If an object has a mass of 2kg and is being pushed by a force of 6N, what is its acceleration (assume the force pushes directly toward the center of mass)?

a) 8m/s2
b) 3m/s2
c) 2m/s2
d) 9m/s2

15. In the following diagram, a force $\mathbf{f} = [-5, 6, -1]$ acts on a rod at point $p = [1, 7, 3]$. Assume the rod rotates freely around the origin. What is the approximate magnitude of the resultant torque vector around the origin?



a) 64.42
b) 23.90
c) 50.02
d) 18.96

Section 2: C/C++

1. What does the following code snippet do (assume that all variables are integers)?

```
nTotal = ++nCurrent;
```

a) Stores the value of nCurrent in nTotal, then increments the value of nCurrent.
b) Adds the value of nCurrent to nTotal.
c) Increments the value of nCurrent, then stores the result in nTotal.
d) Stores the value of nTotal in nCurrent.

16. Could the following enumeration be improved, or is there anything wrong with it?

```
enum
{
        Corn,
        Wheat,
        Barley,
        Rice
};
```

a) "enum" is not the proper C++ keyword to define an enumeration.
b) This enumeration is "anonymous" so the compiler can't do any type-checking on its use.
c) For clarity, the first enumerator should be assigned a value such as 1.0f.
d) You need to separate the enumerators with a semicolon, not a comma.

17. You have a floating point variable and you want to see if it matches one of several potential values. Would a switch statement yield faster run-time performance than a bunch of if-else statements?

a) Yes, because a switch statement is usually implemented with a jump table.
b) No, because a switch statement usually causes cache misses.
c) Yes, because if-else statements wouldn't work in this case
d) No, because a switch statement wouldn't work in this case.

18. In the following code snippet, is there anything you might need to be aware of that isn't obvious from looking at the code?

```
Current++;
```

a) If Current is a class, the copy constructor may be called.
b) If Current is a class, you can't call the post-increment ("++") operator and you can't overload the post-increment operator, either.
c) If Current is an integer, it won't update itself after doing the post-increment.
d) There is nothing that you need to be aware of.

19. What is the following code snippet attempting to do?

```
nResult = nStatus1 & (nStatus2 | nStatus3);
```

a) Take nStatus2 and bitwise-and it with nStatus3. Take that result and bitwise-or it with nStatus1. Take that result and store it in nResult
b) Take nStatus2 and add it to nStatus3. Take that result and multiply it with nStatus1. Take that result and store it in nResult.
c) Take nStatus2 and bitwise-or it with nStatus3. Take that result and bitwise-and it with nStatus1. Take that result and store it in nResult
d) Check if nStatus2 or nStatus3 are the same. If they are, check if they and nStatus1 are the same. If they are, set nResult to be the same.

20. What is the following code snippet attempting to do?

```
nResult = nStatus1 >> nStatus2;
```

a) Take nStatus1 and bitwise-shift-right by the value of nStatus2. Take that result and store in nResult.
b) Check if nStatus1 is greater than nStatus2
c) Take nStatus1 and "pipe" it into nStatus2. Take that result and store in nResult
d) Take nStatus1 and exclusive-or it with nStatus2.

21. You have a program which uses a 32-bit variable (named nAllFlags) to keep track of several flags which are defined in an enumeration. You want to write some code which will return true if a flag is set and false if it is not. Assume that you pass in an integer (named nCode) which contains the enumerated flag you want to check. Which of the following code snippets will best get the job done?

```
a) return ((1 << nCode) & nAllFlags == (1 << nCode));
b) return 1 << nCode & nAllFlags == 1 << nCode;
c) either will work perfectly fine.
d) both could have potential problems.
```

22. In the following code snippet, the value of nTotal ends up being 38 and not 40. Why?

```
int   nValue1 = 12, nTotal = 0;

nTotal += nValue1;

{
      int   nValue1 = 14;

      nTotal += nValue1;
}

nTotal += nValue1;
```

a) nValue1 is initially set to 12. Later, that same variable is set to 14 inside some braces. When the code goes out of scope (leaves the braces), the compiler remembers to restore its value back to 12.
b) nValue1 is initially set to 12. Later, a different variable of the same name is set to 14 inside some braces. When the code goes out of scope (leaves the braces), the original nValue1 is used again.
c) Since we change nValue1 inside some braces, the compiler knows to push its value on the stack upon hitting the opening brace. Upon hitting the closing brace, the compiler pulls the value back off the stack.
d) Trick question: the value of nTotal actually ends up being 40.

23. In the following code snippet, what value will the function return?

```
int   ComputeTotal(void)
{
      int   nTotal, nCount;

      nCount = 10;
      nTotal += nCount;

      nCount--;
      nTotal += nCount;

      return nTotal;
}
```

a) 19
b) 10
c) 1
d) The result is unknown.

24. Under the current ANSI C++ specification, what is wrong with the following code snippet?

```
float fTemp = 0.0f;

for (int j = 0; j < 10; j++)
{
      if (CheckIfDoneEarly(j))
            break;

      fTemp += 2.0;
}

if (j < 10)
      PrintErrorMessage();
```

a) The break statement will not break out of the for-loop since it is associated with the if statement.
b) The variable j is used out of scope.
c) Can't initialize fTemp in the manner shown here.
d) From what is shown here, there is nothing wrong with the function.

25. In the following code snippet, what are the lines attempting to do? Assume that integers are 32-bit, that pnValue is an integer pointer, and that nTotal is an integer.

```
nTotal = *pnValue + 3;
pnValue++;
nTotal += pnValue[3];
```

a) Dereference what pnValue is pointing to, then add three to it. Store that in nTotal.
Increment the address pointed to by pnValue by four bytes.
Dereference the address three integers downstream from what pnValue is pointing to and add to nTotal.

b) Add three to the address pointed to by pnValue and dereference that address and store in nTotal.
Increment the address pointed to by pnValue by four bytes.
Dereference the address three integers downstream from what pnValue is pointing to and add to nTotal.

c) Dereference what pnValue is pointing to, then add three to it. Store that in nTotal.
Increment the address pointed to by pnValue by one byte.
Dereference the address three integers downstream from what pnValue is pointing to and add to nTotal.

d) Dereference what pnValue is pointing to, then add three to it. Store that in nTotal.
Increment the address pointed to by pnValue by four bytes.
Take the address of pnValue, add three to it, and add to nTotal.

26. On some embedded systems that may not have full exception handling capabilities, what is the potential problem with the following code snippet?

```
void  GetValue(void)
{
      CClass      *pcClass;

      pcClass = new CClass[3];

      pcClass[0].m_nMember = 1;
      pcClass[1].m_nMember = 3;
      pcClass[2].m_nMember = 5;

      delete [] pcClass;
}
```

a) pcClass is being de-referenced without first checking if the pointer is valid.
b) You can't create an array of classes.
c) When accessing member variables of pcClass, you should use the "->" operator, not the "." operator.
d) Should not use brackets with the delete operator

27. What is wrong with the following code snippet?

```
int   GetValue(CInput &rcInput)
{
      int   nValue;

      nValue = rcInput.m_nInput1;
      nValue += rcInput.m_nInput2;

      return nValue;
}
```

a) When accessing members of rcInput, you need to use the "->" operator, not the "." operator.
b) This function could have slow execution because the entire contents of the class rcInput has to be pushed on the stack.
c) You cannot pass a reference to a class as an argument to a function.
d) From what is shown here, there is nothing wrong with the function.

28. In the following code snippet, the function GetValue returns 10 and not 17. Why?

```
int   GetValue(void)
{
      int   nValue;

      nValue = 10;
      AddSeven(nValue);

      return nValue;
}

void  AddSeven(int nValue)
{
      nValue += 7;
}
```

a) In the function AddSeven, nValue is passed by value, not by reference, so it does not affect the nValue declared in the function GetValue.
b) The function AddSeven does not return anything (it returns void), therefore, it can't affect anything passed into it.
c) In the function GetValue, the compiler pushes nValue to the stack before calling the function AddSeven and pulls the value after returning. Therefore, the changes made to nValue get lost.
d) Trick question: GetValue actually does return 17.

29. What is wrong with the following code snippet?

```
void  ComputeValue(void)
{
      int   *pnArray, nValue;

      pnArray = new int[10];

      for (nValue = 0; nValue <= 10; nValue++)
            pnArray[nValue] = nValue * 10;
}
```

a) Other than the fact that this function doesn't seem to do anything, there is nothing wrong with it.
b) The function will create a memory leak.
c) The function is writing outside the bounds of an array.
d) Both (b) and (c).

30. What is wrong with the following code snippet?

```
void  Function(void)
{
      int   *pnArray;
      int   nIndex;

      pnArray = new int[10];

      nIndex = 10;

      do
      {
            nIndex--;

            pnArray[nIndex] = nIndex;
      } while (nIndex >= 0);

      delete [] pnArray;
}
```

a) Other than the fact that this function doesn't seem to do anything, there is nothing wrong with it.
b) The function will create a memory leak.
c) The function is writing outside the bounds of an array.
d) Both (b) and (c).

31. What is wrong with the following code snippet?

```
void ComputeValue()
{
      CFoo  *pcFoo1, *pcFoo2;

      pcFoo1 = new CFoo[10];
      pcFoo2 = new CFoo;

      delete pcFoo1;
      delete pcFoo2;
}
```

a) This function is trying to create an array of classes which is not legal in C++.
b) pcFoo2 is not being deleted properly.
c) pcFoo1 is not being deleted properly.
d) Other than the fact that this function doesn't seem to do anything useful, there is nothing wrong with it.

32. What is wrong with the following code snippet?

```
void  CFoo::Function(void)
{
      int   *pnValue;

      pnValue = (int*)malloc(10 * sizeof(int));

      if (pnValue)
            free(pnValue);

      pnValue = new int[10];

      delete [] pnValue;
}
```

a) In the last line, we are not checking if pnValue is non-null before calling "delete".
b) Can't use "malloc" and "free" in a C++ class.
c) In the last line, there should not be any brackets ("[]") after calling "delete".
d) Other than the fact that this function doesn't seem to do anything useful, there is nothing wrong with it.

33. What is wrong with the following code snippet?

```
int*  GetArrayPointer(void)
{
      int   anArray[5] = {10, 20, 30, 40, 50};

      return anArray;
}
```

a) The return statement is missing an ampersand ("return &anArray;").
b) The initializers for anArray are too big. You can't have any values greater than 5.
c) You can't return the address of an array in C++.
d) It is returning a pointer to local data.

34. What do the const keywords do in the following function declaration?

```
void CFoo::Function(const CClass * const pcClass)
```

a) Ensures that neither the contents of what pcClass is pointing to, nor the pointer to pcClass can be modified.
b) You only need the first const since they both say the contents of pcClass can't be modified.
c) Let's the compiler know that the contents of pcClass will be set to a constant value (such as π) in the body of the function.
d) Ensures that pcClass can't modify any members of CFoo, nor will CFoo be able to modify any members of pcClass.

35. If a CClass is derived from CBase, what is the order in which the constructors and destructors will be called?

a) For constructors, first CBase, then CClass. For destructors, first CClass, then CBase.
b) For constructors, first CClass, then CBase. For destructors, first CBase, then CClass.
c) For constructors, first CBase, then CClass. For destructors, first CBase, then CClass.
d) For constructors, first CClass, then CBase. For destructors, first CClass, then CBase's.

36. In the following code snippet the return value of the function GetCount is 15. Why? Assume that this is the first time GetCount has been called and that CFoo has not been previously instantiated.

```
class CFoo
{
public:
      CFoo(){s_nCount += 3;}
      ~CFoo(){}

      static void Add10ToCount(){s_nCount += 10;}

      static int  s_nCount;
};

int CFoo::s_nCount = 0;

int   GetCount(void)
{
      class CFoo  acFoo[5];

      acFoo[0].Add10ToCount();

      return acFoo[4].s_nCount;
}
```

a) The constructor for CFoo adds three to s_nCount, and since there are five instances, the return value is 15.
b) We create five instances of CFoo. We then call Add10ToCount which brings our total count to 15.
c) Trick question: the return value is actually 3 since we are only returning the count from one of the instances of CFoo.
d) Trick question: the return value is actually 25.

37. Which of the following functions would best overload the ">=" relational operator?

a) double CClass::operator >= (CClass &rcClass)
b) void CClass::operator >= (const CClass *pcClass)
c) bool CClass::operator >= (const CClass &rcClass)
d) Trick question: you can't overload the ">=" operator.

38. Is the following class an abstract class?

```
class CFoo
{
public:
      CFoo();
      virtual     ~CFoo();

      void              Function1();
      virtual void      Function2() {};
      virtual void      Function3() = 0;
};
```

a) Yes, because of the function definition for Function2.
b) Yes, because of the function definition for Function3.
c) Yes, because the destructor is declared virtual
d) No, this is not an abstract class.

39. In the following code snippet, how many instances of the function GetValue are created?

```
template<class T> T GetValue(T Input)
{
      T Value;

      Value = Input + Input;

      return Value;
}

void  Function()
{
      int         nInput1 = 1, nOutput1;
      int         nInput2 = 2, nOutput2;
      double      dInput1 = 3.0f, dOutput1;
      double      dInput2 = 4.0, dOutput2;

      nOutput1 = GetValue(nInput1);
      nOutput2 = GetValue(nInput2);
      dOutput1 = GetValue(dInput1);
      dOutput2 = GetValue(dInput2);
}
```

a) 1
b) 2
c) 3
d) 4

Section 3: Code Samples

1. Write a program that converts a hexadecimal (base-16) number into a decimal (base-10) number. The program will take as its input a pointer to a character string which has the hexadecimal number in ASCII. It will return a pointer to a character string which has the decimal number in ASCII. Do not use any standard C libraries (e.g. scanf()).

```
Example:

char* Convert(const char *pHexString)
{
      //do some conversion stuff
      //return result
}
```

40. Write a function that removes a node from a doubly-linked list and deletes it. The structure definition of the linked list is shown below. Fill in the code for the function KillNode().

```
Example:

//linked list node definition
typedef struct LinkNode
{
      struct LinkNode   *pPrev;
      struct LinkNode   *pNext;
      int               *pData;     //dynamically allocated data.
} LinkNode_t

//global pointer to head of linked list
LinkNode_t  *g_pLinkedList;

void KillNode(LinkNode_t *pNodeToRemove)
{
      //do some node removal stuff
}
```