**1. What is the output produced by this program?**

int square(int n)

{

return n\*n;

}

int main()

{

for (int k = 1; k <= 6; k++)

cout << "The square of " << k << " is " << square(k) << endl;

}

**The output of this program is:**

The square of 1 is 1

The square of 2 is 4

The square of 3 is 9

The square of 4 is 16

The square of 5 is 25

The square of 6 is 36

**2. Why doesn't this code compile correctly?**

int square(int n)

{

return n\*n;

}

int main() // This program contains incorrect code!!

{

cout << "Enter an integer: ";

int k;

cin >> k;

cout << "The square of " << k << " is " << square(int k) << endl;

}

This code doesn’t compile correctly because it states “int k” within the square function which is supposed to have an integer argument rather than a declaration of a variable

**3. Why doesn't this code compile correctly?**

int main() // This program contains incorrect code!!

{

int square(int n)

{

return n\*n;

}

cout << "Enter an integer: ";

int k;

cin >> k;

cout << "The square of " << k << " is " << square(k) << endl;

}

This code doesn’t compile correctly because the function is created within the main function whereas it is supposed to be created outside of the main function.

**4. What is the output produced by this program?**

int factorial(int n)

{

int prod = 1;

for (int i = 2; i <= n; i++)

prod \*= i;

return prod;

}

int main()

{

for (int k = 1; k <= 6; k++)

cout << "The factorial of " << k << " is " << factorial(k) << endl;

}

**The output produced by this program is**:

The factorial of 1 is 1

The factorial of 2 is 2

The factorial of 3 is 6

The factorial of 4 is 24

The factorial of 5 is 120

The factorial of 6 is 720

**5. What is the output produced by this program? Between the function printFactorial and the function factorial of the previous question, which is likely to be useful in a wider variety of programs than the other, and why?**

void printFactorial(int n)

{

int prod = 1;

for (int i = 2; i <= n; i++)

prod \*= i;

cout << "The factorial of " << n << " is " << prod << endl;

}

int main()

{

for (int k = 1; k <= 6; k++)

printFactorial(k);

}

The output produced by this program is the same as the output of number 4. The first function, *factorial*, is likely to be more useful than *printFactorial* as you can mix and match the way you want to use the output of *factorial* to whatever situation you want whereas with the *printFactorial* function it can only be used to print out the statement: The factorial of \_\_\_ is \_\_\_

**6. What is the output produced by this program?**

int countChars(string s, char c)

{

int total = 0;

for (int k = 0; k != s.size(); k++)

{

if (s[k] == c)

total++;

}

return total;

}

int main()

{

cout << countChars("abracadabra", 'a') << endl;

int n = 10 \* countChars("99 Cent Only", '9') + 5;

cout << n << endl;

cout << countChars("O Rose, thou art sick!", 'f') << endl;

}

**The output produced by this program is:**

5

25

0

**7. Why doesn't this code compile correctly?**

int countChars(string s, char c)

{

int total = 0;

for (int k = 0; k != s.size(); k++)

{

if (s[k] == c)

total++;

}

return total;

}

int main() // This program contains incorrect code!!

{

cout << countChars('s', "She sells seashells down by the seashore") << endl;

}

This code does not compile correctly because the function specifies the parameters to be a string and then a character in that order but the programmer inputted a character and then a string in that order as the arguments to fulfill the parameters.

**8.What is the output produced by this program?**

bool contains(string s, char c)

{

for (int k = 0; k != s.size(); k++)

{

if (s[k] == c)

return true;

}

return false;

}

int main()

{

if (contains("Computer Science 31", '3'))

cout << "W";

if ( ! contains("Smallberg", 'x'))

cout << "o";

if (contains("Smallberg", 'l'))

cout << "w";

if (contains("Start your CS 31 projects early!", ' '))

cout << "!" << endl;

}

**The output of this program is:**

W

o

w

!

**9. Why does the following program produce the output x instead of xy?**

bool contains(string s, char c) // This code is suspect

{

for (int k = 0; k != s.size(); k++)

{

if (s[k] == c)

return true;

else

return false;

}

}

int main()

{

if (contains("xyz", 'x'))

cout << "x";

if (contains("xyz", 'y'))

cout << "y";

cout << endl;

}

The program outputs *x* instead of *xy* because the code for the function *contains* has an *else* function after the *if* statement which returns false which means that if the character specified in *c* is not the first character of the string *s*, the *if* statement in the *main* function will evaluate to false for the second *if* statement, as it did in this program.

**10. Why does the following program exhibit undefined behavior (i.e., you might get different results on different machines)?**

int countSpacesInFirstSentence(string s) // This code is suspect

{

int nSpaces = 0;

for (int k = 0; k != s.size(); k++)

{

if (s[k] == ' ')

nSpaces++;

else if (s[k] == '.')

return nSpaces;

}

}

int main()

{

cout << countSpacesInFirstSentence("Call me Ishmael.") << endl;

cout << countSpacesInFirstSentence("I am Sam. Sam I am.") << endl;

cout << countSpacesInFirstSentence("Where now? Who now? When now?") << endl;

}

This program exhibits undefined behavior because when there are no ‘ ’ characters and no ‘.’ characters in the string there is no path for the function to take and thus the program will spit out a “garbage” value.

**11. What is the output of the following program?**

string justLetters(string s)

{

string result = "";

for (int k = 0; k != s.size(); k++)

{

if ( islower(s[k]) )

result += s[k];

if ( isupper(s[k]) )

result += tolower(s[k]);

}

return result;

}

int main()

{

cout << justLetters("CS 31 began September 27.") << endl;

}

**The output of the following program is:**

csbeganseptember