

## 1 c++ and Errors

Complete the following two tasks for each of the following code snippets:

1. Circle the line(s) that cause an error.
2. Categorize each of the following code snippets by the type of error that they produce: runtime, compile time, or no error.
3. You may assume all needed libraries have been `#included`.

```
1 int main() {  
2     int a = 10;  
3     std::string b = "cat";  
4     std::cout << (a + b) << std::endl;  
5 }
```

```
2 int main() {  
2     int a = 10;  
3     std::string b = "cat";  
4     std::cout << a << b << std::endl;  
5 }
```

```
3 void PrintContents(std::vector<int> v) {  
2     for (int i = 0; i <= v.size(); i++) {  
3         std::cout << v[i] << std::endl;  
4     }  
5 }  
6  
7 int main(int argc, char* argv[]) {  
8     std::vector<int> v = {1, 2, 4};  
9     PrintContents(v);  
10 }
```

```
4 struct Book {  
2     std::string title;  
3 };  
4  
5 void PrintContents(std::vector<Book> v) {  
6     for (int i = 0; i < v.size(); i++) {  
7         std::cout << v[i].title << std::endl;  
8     }  
9 }  
10  
11 int main(int argc, char* argv[]) {  
12     Book b;  
13     b.title = "BFG";  
14     std::vector<Book> v = {b};  
15     PrintContents(v);  
16 }
```

```
5 int main(int argc, char **argv) {  
2     std::cout << argv[0] << std::endl;  
3     std::cout << argv[1] << std::endl;  
4 }
```

- 5b. What type is argv in the above code snippet?

Name: \_\_\_\_\_

## 2 Static type checking

1. When does static type checking happen?
2. What are at least 3 specific benefits of static type checking?

### 3 Python and errors

Useful tips for python:

`print(var1, var2)` is equivalent to `cout << var1 << " " << var2 << endl;`.

`range(number)` produces a list of integers from 0 to `number - 1`.

In python 3, `"/"` is float divide and `"//"` is integer divide.

```
1 def main():
2     a = 10
3     b = "cat"
4     print(a + b)
5
6 main()
```

```
2 def main():
2     a = 10
3     b = "cat"
4     print(a, b)
5
6 main()
```

```
3 def print_list(ls):
2     for i in range(len(ls) + 1):
3         print(ls[i])
4
5 def main():
6     ls = [1, 2, 4]
7     print_list(ls)
8
9 main()
```

```
4 def print_list(ls):
2     for i in range(len(ls)):
3         print(ls[i])
4
5 def main():
6     ls = ["cat", 1236, True, False, 0.123]
7     print_list(ls)
8
9 main()
```

```
5 import sys
2
3 def main():
4     print(sys.argv[0])
5     print(sys.argv[1])
6
7 main()
```

```
6 def main():
2     for i in range(10):
3         print("Hello, world!")
4
5 main()
```

Name: \_\_\_\_\_

## 4 add\_to\_values

```
1 def add_to_values(ls, v):  
2     for i in range(len(ls)):  
3         ls[i] = ls[i] + v
```

1. Given the above function definition, write down 6 function calls to `add_to_values`, all with the correct number of parameters and that use a list or a string as values for the first parameter. Which of them produce errors? Make sure **at least** 2 of the function calls produce errors.

## 5 Dynamic type checking

1. When does dynamic type checking happen?
2. What are at least 3 specific benefits of dynamic type checking?