

Lab 10 Report

Basics of Programming 2

Das Anjan Kumar

D42DQA

Task B.

Put breakpoints in all constructors and conversion operators; then run the program in Debug mode. Observe the constructor and conversion operator calls. Explain.

Answer:

The constructors and conversion operator calls are as followed.

Before the output of first line, first the constructor to covert character pointer to string is called, then gets a read-only access to String and then print.

```
String(const char* str)
operator const char* () const
```

Hello, Mrs. Moneypenny.

Before the second line output, firstly again the covert character array to string is called, then assigning string, then again character array conversion constructor , then add assignment operator, then the default constructor, and string argument constructor, and finally assignment operator with reading string is called,

```
String(const char* str)
const String& operator=(const String& theOther)
String(const char* str)
const String& operator+=(const String& theOther)
String()
String(const String& string)
```

```
const String& operator=(const String& theOther)
operator const char* () const
```

My name is Bond. James Bond.

For the final line output, as before, first the character array constructor call then comparing, then another character array constructor call and comparing and finally the output.

```
String(const char* str)
bool operator==(const String& theOther)const
String(const char* str)
bool operator!=(const String& theOther)const
```

Oh, Mr. Bond!

Task c) If we leave out the explicit type conversion from the following piece of code, then we shall get an error message:

```
str1 == (String)"My name is Bond. James Bond.";
```

What kind of error is that? Try to find the cause of the error.

Answer:

We get this error:

Severity	Code	Description	Project	File	Line	Suppression State
Error (active)	E0350	more than one operator "==" matches these operands:	String	C:\Users\Anjan\Desktop\BOP 2\String\String\TheUltimateString.cpp	16	

The reason of this is, conflict between actual == operator and operator== overloaded. That's why we need to explicitly declare string, so that it matches with the overloaded operator call.