TATYANA VLASKIN
WEEK 9
PROBLEM 1:

1. We need to create a class
2. Within a class, we need to create a public variable name and public variable age
3. We need to create a function that creates an object of the class and is not a member of the class
4. The function will ask the user to enter name and age
5. We also need a function that will check and make sure that entry made for age is valid
6. Name and age will be echoed back to the user.

CLASS:

```
class person // newperson is the tag
{
public:
  char name[20]
  int age
     };
```

FUNCTIONS (NOT PART OF THE CLASS):

```
person new_person();// function that will create object person person1 and will ask for name and age
// this function will be basic cout/cin
// to echo name and age back to the user:
   cout <<  "Your name is: " << person1.name << endl;
   cout << "You are " << person1.age << endl;
void integerCheck(int &age);// function that will check to make sure that integer was entire for age
```

PROBLEM 2:

This looks like a continuation of the problem 1.
We need to create function that will be defined as a part of the class.
The function will be called **having_birthday()**.
The function will increment the age of the person.
I will reuse code from problem 1, with few modifications.
Ill add formal parameter to function: person new_person(person & person1);
Ill add function to the class: void having_birthday(person & person1);

```
void person::having_birthday(person & person1)
{
   age = age + 1;
   cout << "Next year you will turn " << person1.age << endl;
}
```

PROBLEM 3:
For this problem, we need to take previous problem and change the member variables of name and age to be private.
We need to add “accessor” member functions for the name and age. These functions will ask user to input name and age.

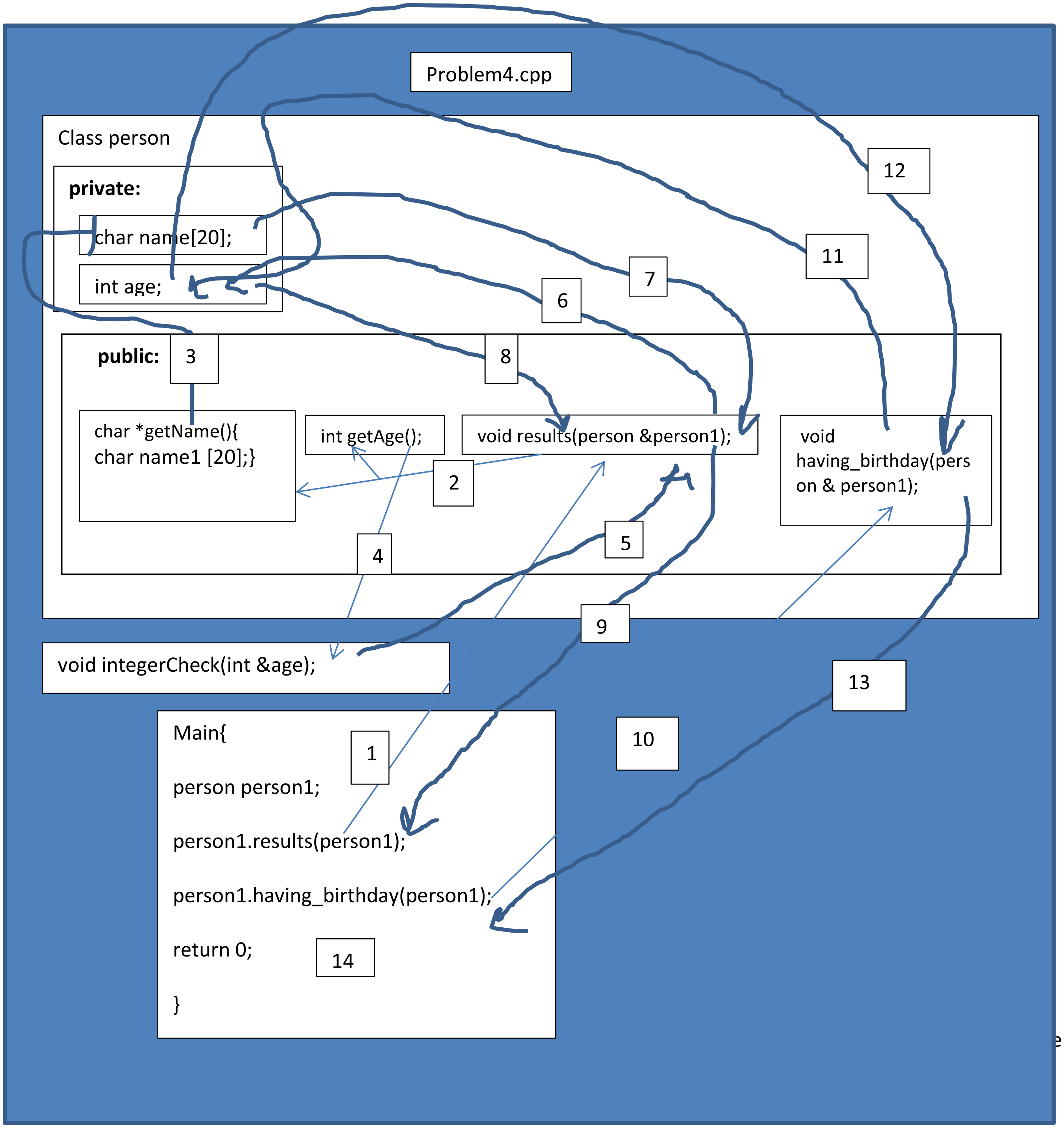
So the class will be changed to:

```
class person // newperson is the tag
{
   private:
      char name[20];// member 1
      int age;// member 2
   public:
      char *getName();// function will ask user to enter name
      int getAge();// function will ask user to enter age
      void results(person &person1);// function will display age and name back to the
user
      void having_birthday(person & person1);// function will increment age by 1
};
```

For the function definition, I’ll be doing copy and paste of the code from the problem 2.

```
//function to enter name
char *person::getName(){
   char name1 [20];
   cout << "What is your name? ";
        cin.get (name1, 20);
   cin.ignore(80,'\n');
   strcpy (name,name1);
   return name;
}
//function asks the user enter age
int person::getAge(){
   cout << "How old are you? ";
        cin >> age;
        integerCheck(age);
   return age;
}
//function displays age and name back to the user
void person::results(person &person1){
    cout << "Your name is: " << person1.getName() << endl;
    cout << "You are " << person1.getAge() << endl;
}
//function increments age by 1
void person::having_birthday(person & person1)
{
    age = age + 1;
    cout << "Next year you will turn " << person1.age;
}
```

PROBLEM 4:



Flow of the program:

Step 0: program is started and person person1 object is created

Step 1: person1.results(person1) function is called

Step 2: char *getName() and int getAge are called

Step 3: User enters a name and name is stored in the private variable name

Step 4: User enters age and void integerCheck(int &age); function is called to check if entry is valid.

Step 5: If entry is not valid, the user is asked to reenter age. If entry is valid, its accepted

Step 6: Age is astorered as a private variable age

Step 7 and 8: Variables name and age are accessed by functions person1.results(person1)

Step 9: person1.results(person1) return name and age values in the main method

Step 10: person1.having_birthday(person1); is called

Step 11: person1.having_birthday(person1); accesses private variable age

Step 12: person1.having_birthday(person1) increments age by 1

Step 13: incremented age is returned in the main function

Dtep 14: return 0 is the end of program

The main difference between classes and structure is that by default variable are private in classes and public in structures.

PROBLEM 5:

1. This is continuation of problem 3, so code will be reused
2. We need to create a variable to store an object
3. We need to create a pointer that point to the object
4. Access the address of the object
5. Access the members through the pointer
6. We also need to create an array of objects
7. Fill the array with variables
8. Access the member variable values in the array

   DESIGN:

   Ill split getName() function into 2 functions: 1 will get name from the user and the second one will return name.

   Same thing will be done for the age as well. This is necessary because most likely Ill be accessing age and name several time, so I do not want to constantly ask user to reenter name and age.

So, new class will look like this:

```
class person
{
   private:
      char name[20];
      int age;
   public:
      void *setName();
      char *getName();
      void setAge();
      int getAge();
      void results(person &person1);
      };
```

This program will involve a lot of copy and paste and rearrangement of problem 3.

```
void *person::setName(){
   char name1 [20];
   cout << "What is your name? ";
  cin.get (name1, 20);
   cin.ignore(80,'\n');
   strcpy (name,name1);
}

char *person::getName(){
    return name;
}
```

Before return name was part of setName()

```
void person::setAge(){
   cout << "How old are you? ";
    cin >> age;
    integerCheck(age);
   }

int person::getAge(){
   return age;
}
```

Before return age was part of setAge() function

Function void person::results(person &person1 will be rewritten as:

```
void person::results(person &person1){
    person1.setName();
    cout << "Your name is: " << person1.getName() << endl;
    person1.setAge();
    cout << "You are " << person1.getAge() << endl;
}
```

Object person person1 will be created.

person1.results(person1) function will be called

pointer person *pointerPerson; will be created

pointerPerson = &person1- the pointer will point to the memory address of the object person1.

Next there will be several cout statements:

1. "The memory address to the class person is: "<<pointerPerson
2. Value stored in the member name can be accessed as: pointerPerson->getName()
3. Value stored in the member name can be accessed as: (*pointerPerson).getName()
4. Value stored in the member age can be accessed as: pointerPerson->getAge()
5. Value stored in the member age can be accessed as: (*pointerPerson).getAge()

Next, Ill create array of object and fill in array with values (names and ages)

```
 cout<<"Enter names and ages of 3 people who are nominated to the best TA
award."<<endl;
    person bestTA[3]; // ARRAY OF OBJECTS
    person *pointerBestTA[3]; //ARRAY OF POINTERS
```

Initially my code looked like this:

```
    cout<<"Enter names and ages of 3 people who are nominated to the best TA
award."<<endl;
    person bestTA[3];
    person *pointerBestTA[3];
    for(int i=0; i<3;i++){
        cout <<"What is the name of the " << i+1 << " TA? ";
        bestTA[i].setName();
        pointerBestTA[i] = &bestTA[i];
        cout << "How old is the " << i+1 << " TA ? ";
        bestTA[i].setAge();
      }
```

However, the program was not letting me enter name and age of the second person.

I always have issues with cin/ cin.get and it seems that cin.ignore should fix the problem, but it NEVER does and each time, I have to play with the code to make it work. I do not get it. This time, I decided to add two more functions to the class person: void *setNameTA(); and void setAgeTA(); . I am also adding newline function discussed in the book. This fixed the problem and now I can enter all the names and ages.

```
void newLine( ){
   char symbol;
   do
   {
      cin.get(symbol);
   } while (symbol != '\n');
```

This function will continue reading until it encounters new line

```
void person::setAgeTA(){
    cin >> age;
    integerCheck(age);
   }
```

cout << "How old are you? "; is removed from the setAge () fuction.

```
void *person::setNameTA(){
   char name1 [20];
   cin >> name1;
    newLine();
   strcpy (name,name1);
}
```

cin.get (name1, 20);
is change in cin and newline function is added instead of cin.ignore
Also the question what is your name is removed from the function

To fill in the array of object and array of pointers, the following loop will be used:

```
for(int i=0; i<3;i++){
      cout <<"What is the name of the " << i+1 << " TA? ";
      bestTA[i].setNameTA();//array of objects is filled
      pointerBestTA[i] = &bestTA[i];// array of pointers is filled
      cout << "How old is the " << i+1 << " TA ? ";
      bestTA[i].setAgeTA();//array of objects is filled
     }
```

Once the array is filled, the following code will be used to access members in the array and to display the addresses to which array of pointes is pointing.

```
cout << "ARRAY OF OBJECTS: " << endl;
   cout <<"Accessing member variable values in the  person bestTA[3] array "
      << endl << "using dot (.) operator. " << endl;
   for(int i=0; i<3;i++){
      cout << i+1 << " TA, " << bestTA[i].getName()
          << ", is " << bestTA[i].getAge() << " year old "  << endl;
   }
   cout << "ARRAY OF POINTERS: " << endl;
   cout <<"Accessing member variable values in the  person bestTA[3] array "
        << endl <<"using arrow ( -> ) operator. " << endl;
   for(int i=0; i<3;i++){
      cout << i+1 << " TA, " << pointerBestTA[i] -> getName()
          << ", is " << pointerBestTA[i] ->getAge() << " year old "  << endl;
      cout<<"Pointer at index "<<i <<" of the array of pointers points to memory
address: "<<pointerBestTA[i]  <<endl;
   }
```